

# 开发月刊

Development Monthly

2011年11月

总第008期



Celebrating 10 years of Eclipse

Eclipse十年记

Node.js提速指南

死诸葛吓走生仲达？移动之战刚开打

## 编程排行 Billboard

- 3 11月编程排行榜:Google新利器Dart

## 专题报道 《Eclipse十年庆》

- 6 Eclipse十年磨一剑，霜刃未曾试  
8 Eclipse推JVM语言Xtend 拥抱Java  
9 Eclipse调试Bug的七种常用技巧

## 技术热点 Techlogy hot

- 11 程序员们 不要想一辈子靠技术混饭吃  
12 10月Java备忘录：盘点Java主要成就  
14 Scala难在哪里？  
16 项目经理不可不知的“三边六拍”  
18 10月Web技术最前沿:JavaFX 2发布  
20 Node.js提速指南  
21 为什么我如此迷恋Lisp语言？  
23 揭秘Twitter网站架构

## 移动开发 Mobile Developpe

- 25 死诸葛吓走生仲达？移动之战刚开打  
27 美女与极客之战:Mango VS Android  
30 李洋专栏：Android应用签名概述  
32 杨亨冉Silverlight实现跨平台游戏开发  
34 挑战者诺基亚：放下辉煌，从头再来  
36 Adobe的光棍节:与移动版Flash player离婚之后

## 特别推荐 Best topic

- 38 51CTO邀您参加Phone Club技术沙龙



# 11月编程排行榜:Google新利器Dart

离 2012 年只有两个月了,从目前有趋势来看, Objective-C 已经脱颖而出成为最佳编程语言奖项的最有力竞争者,本月更上升到排行榜第六位,另外 C#, JavaScript 也有所上升,2010 年和 2007 年两次获得年度编程语言奖项的 Python 排名却有所下滑,不过仍然排行第 8 位。

本月 51CTO 为您介绍 Google 新结构化编程语言 Dart。早在九月份 Google 就已着手注册了一系列与 Dart 相关的域名,当时也引发了沸沸扬扬的猜测。只到 10 月 10 日在丹麦举行的 Goto 会议上,谷歌 Dart 语言项目的领导人 Lars Bak 才正式宣布推出编程语言 Dart。Lars Bak——这位来自谷歌的工程师曾经凭借其 V8 JavaScript 解释程序颠覆了火狐浏览器的速度神话。

下面是前 20 名的编程语言排行

Position Nov 2011	Position Nov 2010	Delta in Position	Programming Language	Ratings Nov 2011
1	1	=	Java	17.874%
2	2	=	C	17.322%
3	3	=	C++	8.084%
4	5	↑	C#	7.319%
5	4	↓	PHP	6.096%
6	8	↑↑	Objective-C	5.983%
7	7	=	(Visual) Basic	5.041%
8	6	↓↓	Python	3.617%
9	11	↑↑	JavaScript	2.565%
10	9	↓	Perl	2.078%
11	10	↓	Ruby	1.502%
12	20	↑↑↑↑↑↑	PL/SQL	1.438%
13	13	=	Lisp	1.182%
14	15	↑	Pascal	0.991%
15	21	↑↑↑↑↑	MATLAB	0.955%
16	12	↓↓↓	Delphi/Object Pascal	0.872%
17	23	↑↑↑↑↑	ABAP	0.847%
18	22	↑↑↑	Lua	0.635%
19	16	↓↓↓	Ada	0.622%
20	19	↓	RPG (OS/400)	0.620%

## 什么是 Dart?

Dart 是一种基于类的可选类型化编程语言,设计用于创建 Web 应用程序。Dart 的设计目标是为 Web 编程创造结构化但又富有灵活性的语言;编程方法一目了然,符合程序员的自然习惯,易于学习;能在所有浏览器 和不同环境中实现高性能。谷歌推出的全新 Web 开发语言 Dart 很可能与 Smalltalk 存在着种种相似之处。它将 Web 发展引向一条与此前截然不同的另类道路。参考 51CTO 外电:

Dart 之于 JavaScript 正如 C# 之于 C++

## Dart 语言 “Hello world”

```
class HelloDartTest {  
    static testMain() {        print("Hello, world!");  
} } main() {    HelloDartTest.testMain(); }
```

## Dart 如何执行代码

Dart 主要有两种执行方式:

1. 在浏览器中执行。

a) 在 Dartboard 中,在线编辑、执行,尝试和演示目的。这是目前最简单的方法。

b) 使用转换工具将 dart 转换为 javascript 后在浏览器执行。

c) 使用 script 标签在浏览器中直接执行,。当然浏览器要支持才行,目前还没有,估计只有 Chrome 将会支持。



## 11 月编程语言排行榜 :Google 新利器 Dart

2. 虚拟机直接执行。dart 源文件的后缀一般是 .dart ,就像其它语言一样: dart\_bin yourcode.dart。

默认执行不启用类型检查,可以使用 `--enable_type_checks` 选项启用类型检查。

目前官方还没提供个平台的二进制构建版本,需要自行获取代码构建。

或网上下载别人构建好的版本,具体见这里, Linux、Win、Mac 都有 (没试过不保证没问题)。

### Dart 设计目标

Dart 编程语言当前处于早期阶段。这些设计目标将会持续地演进和精炼:

- ◆ 创建一种结构化但灵活的 web 编程语言
- ◆ 使用程序员感到熟悉并自然,易于学习
- ◆ 确保所有的 Dart 语言构造允许高性能和快速启动应用

- ◆ 使 Dart 适用于全部范围的 web 设备,包括手机、平板、笔记本和服务端

- ◆ 提供工具使 Dart 能够在所有现代浏览器上快速运行

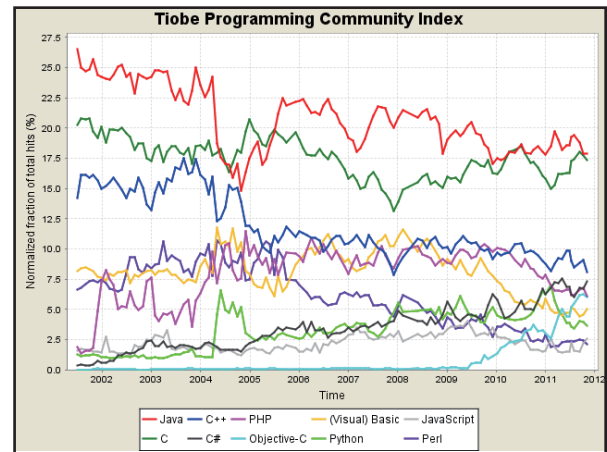
这些设计目标定位于当前 Web 开发者面对的如下问题:

- ◆ 小脚本经常演化为没有明显结构的大型 web 应用,他们难以调试和维护。另外,这些整体的应用不能被切分所以不同小组不能独立的工作。当 web 应用变大之后很难有生产力。

- ◆ 脚本语言的流行是因为他们轻量型的本质使其快速写代码很容易。通常应用与其它部分的契约是写在注释中的而不是在语言自身的构造中。结果,作者以外的人难以阅读和维护一段特定的代码。

本期编程语言排行榜的其他排名数据和趋势走向。

前 10 名 编程 语言 走 势 图



下面是第 50 到 100 的编程语言排名

(Visual) FoxPro, ABC, Algol, Bash, bc, BETA, BlitzMax, Boo, Bourne shell, CFML, cg, CL (OS/400), Clean, cT, Dart, Dylan, Eiffel, Factor, Free Pascal, Groovy, Icon, IDL, Io, J, JavaFX Script, JScript.NET, Korn shell, LabVIEW, Lingo, LPC, MAD, MUMPS, NATURAL, Oberon, Occam, OpenCL, OpenEdge ABL, Oz, PostScript, PowerShell, REXX, S, Scala, SPSS, Standard ML, SuperCollider, VBScript, X10, xBase, Z shell

下面给出了编程语言类别的一年变化趋势

Category	Ratings Nov 2011	Delta Nov 2010
Object-Oriented Languages	55.3%	-0.5%
Procedural Languages	38.1%	-0.6%
Functional Languages	4.3%	+0.5%
Logical Languages	2.3%	+0.6%

Category	Ratings Nov 2011	Delta Nov 2010
Statically Typed Languages	63.4%	+0.5%
Dynamically Typed Languages	36.6%	-0.5%

更多内容,请查看原文,链接:

<http://developer.51cto.com/art/201111/300787.htm> ■



Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具(Java Development Tools, JDT)。

从 2001 年 IBM 将 Eclipse 贡献给开源社区已经过去十年。这十年是 Eclipse 努力向上的十年，也见证了 Java 社区十年来的发展。我们将帮助广大技术人回顾 Eclipse 十年来的发展，更好的理解 Eclipse 的技术理念。

Eclipse 相关专题：

Eclipse 十年庆

<http://developer.51cto.com/developer/Eclipse10/>

51CTO 开发频道寄语

# Eclipse十年磨一剑，霜刃未曾试

作者 / 彭凡

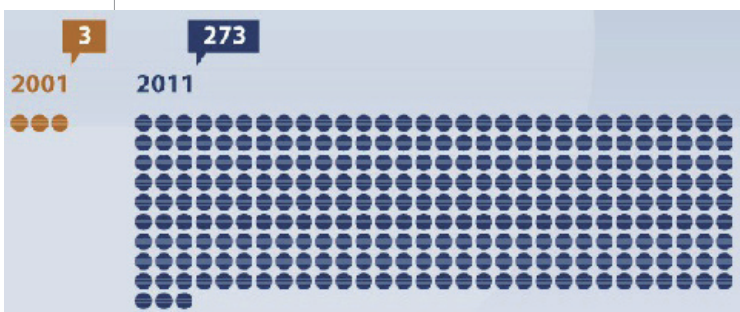
十年时间,可以让乔布斯透过 iPad 和 iPhone 成就苹果的再次崛起,也可以让盖茨的 Windows XP 主导 PC 桌面一个时代。而对于 Eclipse 来说,十年是伴随 Java 阵营逐步壮大的过程。

## 2001 年 11 月 Eclipse 诞生

当 IBM 把带有 Visual Age for Java 基因的 Eclipse 捐献给开源社区的时候,肯定是对 Eclipse 给予了更大的希望。很多人可能记住了作为开发环境 IDE 存在的 Eclipse,而忘记了它是一个开放源代码的、基于 Java 的可扩展开发平台。

2001 年 11 月 7 日, Eclipse 1.0 诞生。到现在各位 Java 程序员还可以在官网下载到这个版本的 SDK,地址是:

<http://archive.eclipse.org/eclipse/downloads/drops/R-1.0-200111070001/index.php>。大家可以自己下载,回忆一下当初的开发感觉。



## 十年间 Eclipse 上运行的项目数目对比图

到 2011 年 8 月 6 日, Eclipse 基金会推出了 Eclipse 3.8M1 以支持 Oracle 最新发布的 Java 7。

## 一年生聚,三年教训

2001 年 Eclipse 1.0 诞生后,到 2003 年已经发展到 Eclipse 2.1。虽然这个版本大受欢迎,但它的插件模型仍然是静态的。

三年后的 2004 年, Eclipse 3.0 正式发布,成为一个完美的、可扩展的开发环境。此时 Java 开发人员已经达到 300 万。2005 年 6 月, Eclipse 3.1 发布,30 天达到 100 万下载量。而此时全世界有大约 450 万名开发者正在使用 Java 技术。



## Eclipse 十年来代码行数从 150 万到 5000 万 +

上面的新闻报道,是 Eclipse 前五年的发展过程。从诞生时的小有瑕疵,到后来的大红大紫。Eclipse 成为众多 Java 开发者的首选。当年小编在大学学习 Java 开发时,用到的环境就是 Eclipse+Tomcat 进行开发。IDE 在代码写完后,很方便的就能直接编译并显示结果,效率上比一开始用命令行运行 J2EE 1.5 要高。

再后来, Eclipse 就逐渐坐稳了 Java 主流编译环境的位子。但似乎隐约中还缺少着什么,对未来还有一丝不确定。

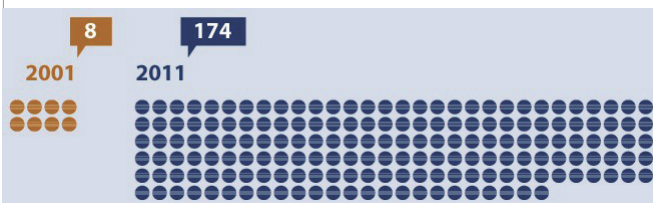
## 十年剑已成,新出鞘

Eclipse 经历的这十年,是 IT 产业从 PC 软件客户端向 WEB 互联网应用发展的十年。随着 AMD、微软、Adobe 等公司的支持加大, Eclipse 也获得了 JCP 的认可成为其中的一员。尽管之前 Sun 一直不同意 Eclipse 加入 JCP,以避免其冲击 Netbeans 的地位。

## Eclipse 十年磨一剑,霜刃未曾试 II

但现在看来两者都在共同支撑着 Java 阵营的前进,与微软 Visual Studio 平台分庭抗礼。

MyEclipse 作为 Eclipse 的补充,在 2011 年也进化到 MyEclipse 10 版本。不知道是巧合还是故意安排,这个版本号与 Eclipse 十年不谋而合。这些最新版本的开发环境都已经支持 Java 7。



### 十年间 Eclipse 合作公司数量对比

前面我们提到了这十年是 PC 到 WEB 的进化历史,同时也是开发界从 WEB 到移动开发的转折点。随着 iOS 和 Andriod 的兴起,更多的传统开发者正在从软件开发转向移动开发。预计在 2012 年将会出现移动开发人员井喷式增长。

作为已经磨砺十年的 Eclipse,在 J2ME 时代已经可以通过 Nokia Developer's Suite for J2ME 这样的工具为 NOKIA 开发移动应用程序。而在 Android 时代,MyEclipse 也同样可以写出优良的 Android 应用。可以说 Eclipse 下一步发展方向一定不要忽视移动开发,特别是基于 Java 的移动应用开发。利剑出鞘会有时!

以上数据均来自 Eclipse 官网

### Eclipse 最全快捷键 分享便捷与快乐

Ctrl+1 快速修复 (最经典的快捷键,就不用多说了)

Ctrl+D: 删除当前行

Ctrl+Alt+ ↓ 复制当前行到下一行 (复制增

加)

Ctrl+Alt+ ↑ 复制当前行到上一行 (复制增加)

Alt+ ↓ 当前行和下面一行交互位置 (特别实用,可以省去先剪切,再粘贴了)

Alt+ ↑ 当前行和上面一行交互位置 (同上)

Alt+ ← 前一个编辑的页面

Alt+ → 下一个编辑的页面 (当然是针对上面那条来说了)

Alt+Enter 显示当前选择资源 (工程,or 文件或文件) 的属性

Shift+Enter 在当前行的下一行插入空行 (这时鼠标可以在当前行的任一位置,不一定是最后)

Shift+Ctrl+Enter 在当前行插入空行 (原理同上条)

Ctrl+Q 定位到最后编辑的地方

Ctrl+L 定位在某行 (对于程序超过 100 的人就有福音了)

Ctrl+M 最大化当前的 Edit 或 View (再按则反之)

Ctrl+ / 注释当前行,再按则取消注释

Ctrl+O 快速显示 OutLine

Ctrl+T 快速显示当前类的继承结构

Ctrl+W 关闭当前 Editor

Ctrl+K 参照选中 Word 快速定位到下一个

Ctrl+E 快速显示当前 Editor 的下拉列表 (如果当前页面没有显示的用黑体表示)

Ctrl+/(小键盘) 折叠当前类中的所有代码

Ctrl+ × (小键盘) 展开当前类中的所有代码

全部: <http://developer.51cto.com/art/201105/264710.htm>



# Eclipse推JVM语言Xtend 拥抱Java

作者 / 彭凡

Eclipse 宣称 Xtend 并无意替代 Java,而是补 Java 之短,精简代码,无类型,改进可读和维护。Eclipse Xtend 可以编译成可读的 Java 代码,类似 CoffeeScript 之于 Javascript。

下载地址:

<http://www.eclipse.org/Xtext/xtend/#download>

## 用 Xtend 编写 Java 代码

在 Xtend 中,可以直接得到 Java 代码。并且你可以在 Eclipse 中很直观的看到这些代码,即使你是从 Java 中进行调用的,你也能看到 Xtend 代码。

## Xtend 支持 Eclipse 工具

Xtend 将支持 Eclipse 上的 Java 开发工具,另外的功能包括

- ◆ 语法着色
- ◆ 内容辅助
- ◆ 重命名重构
- ◆ 组织进口
- ◆ 快速修复
- ◆ 丰富的悬停
- ◆ 大纲视图
- ◆ 导航
- ◆ 开放式

... .. 和许多

## Java 类型系统

静态类型是不错的,因为它允许更好的静态分析和基于类型信息的更好的工具。然而,缺点是额外的复杂性(静态)类型系统。

Java 类型系统尽管众人皆知,但看起来似乎还不太完美。

Xtend 将重用能用到的一切。

```
// qualified names
java.lang.Object
// primitives
boolean, int, long, char, ...
// arrays
String[]
// generics
List<? extends CharSequence>
java.util.Map<String,String>
```

不同的类型判断

变量声明

```
val names = getTheListOfNames()
val List<String> names =
getTheListOfNames()For-Loops
```

for 循环

```
for (name : getTheListOfNames())
// do something with name
for (String name : getTheListOfNames())
// do something with name
```

返回类型

```
def getTheListOfNames() {
    newArrayList("Tomte","Pippi","Carlson")
}
def List<String> getTheListOfNames() {
    newArrayList("Tomte","Pippi","Carlson")
}
```

Closures

```
getTheListOfNames().map( name | "Mr.
"+name )
getTheListOfNames().map( String name |
"Mr. "+name )
```

原文:

<http://developer.51cto.com/art/201111/300828.htm>



# Eclipse调试Bug的七种常用技巧

本文写给那些像几年前的我一样刚刚走出校门,及一些未使用过这些高级些的调试技巧的人。



记得刚刚毕业的时候,自己连断点也不会打,当时还在用JCreate,就连毕业设计也是用System.out找Bug的,想想真的很笨。开始工作后,一个星期过去了,在一个1、2百万行的系统中找Bug,我依然在用System.out,当时最痛苦的就是修改代码,每次找到疑似Bug,就输出一下,然后重启(那时也不知道代码热替换),直到有一天带我的导师发现了这样笨笨的调试Bug,才让我第一次认识了断点,也知道了代码修改完了可以进行热替换,我这个中国教育的半牺牲品才算向美好生活迈进了一小步。

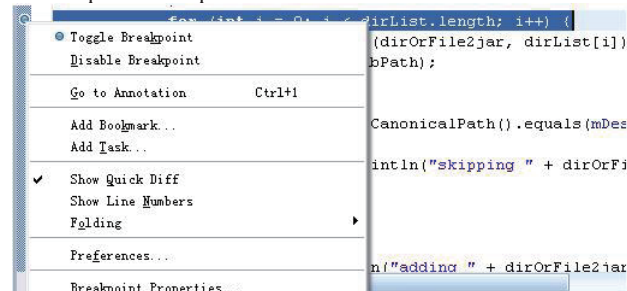
## 1、条件断点

断点大家都比较熟悉,在Eclipse Java编辑区的行头双击就会得到一个断点,代码会运行到此处时停止。

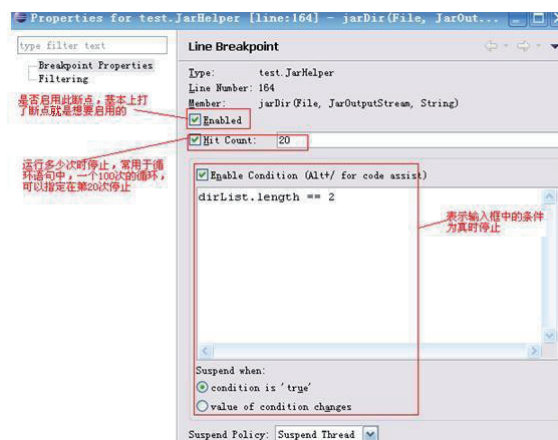
条件断点,顾名思义就是一个有一定条件的断点,只有满足了用户设置的条件,代码才会在运行到断点处时停止。

在断点处点击鼠标右键,选择最后一个

"Breakpoint Properties"

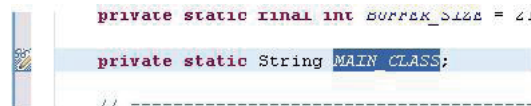


断点的属性界面及各个选项的意思如下图,



## 2、变量断点

断点不仅能打在语句上,变量也可接受断点



上图就是一个变量的打的断点,在变量的值初始化,或是变量值改变时可以停止,当然变量断点上也是可以加条件的,和上面的介绍的条件断点的设置是一样的。

## 3、方法断点

方法断点就是将断点打在方法的入口处



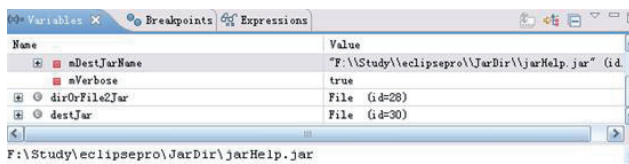
方法断点的特别之处在于它可以打在JDK的源码里,由于JDK在编译时去掉了调试信息,所以普通断点是不能打到里面的,但是方法断点却可以,可以通过这种方法查看方法的调用栈。

## Eclipse 调试 Bug 的七种常用技巧 II

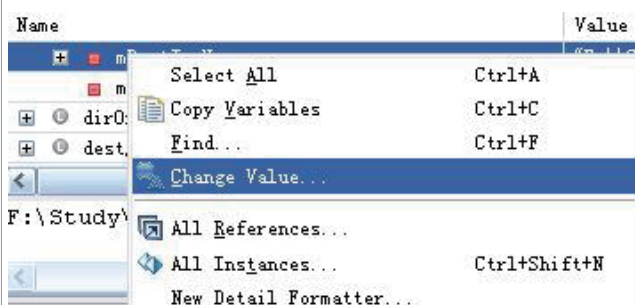
## 4、改变变量值

代码停在了断点处,但是传过来的值不正确,如何修改一下变量值保证代码继续走正确的流程,或是说有一个异常分支老是进不去,能不能调试时改一下条件,看一下异常分支代码是否正确?

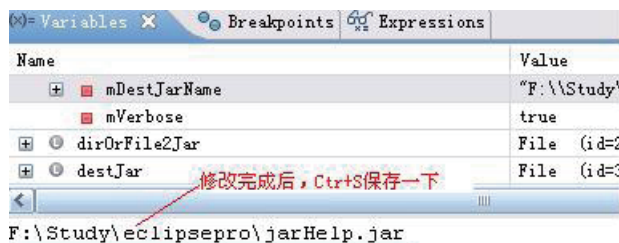
在 Debug 视图的 Variables 小窗口中,我们可以看到 mDestJarName 变量的值为 "F:\Study\ eclipsepro\JarDir\jarHelp.jar "



我们可以在变量上右键,选择 "Change Value..." 在弹出的对话框中修改变量的值



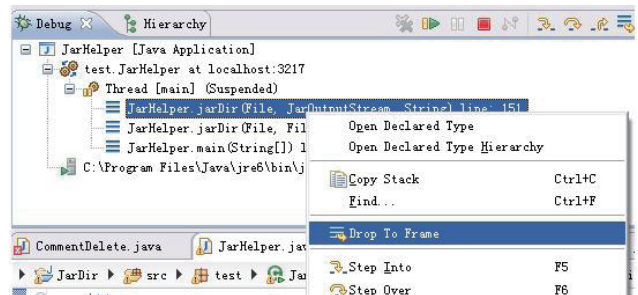
或是在下面的值查看窗口中修改,保用 Ctrl+S 保存后,变量值就会变成修改后的新值了。



## 5、重新调试

这种调试的回退不是万能的,只能在当前线程的栈帧中回退,也就是说最多只能退回到当前线程的调用的开始处。

回退时,请在需要回退的线程方法上点右键,选择 "Drop to Frame"



## 6、异常断点

经常遇见一些异常,然后程序就退出来了,要找到异常发生的地方就比较难了,还好可以打一个异常断点



上图中我们增加了一个 NullPointerException 的异常断点,当异常发生时,代码会停在异常发生处,定位问题时应该比较有帮助。

## 7. 远程调试

用于调试不在本机上的程序,有两种方式,

- 1、本机作为客户端
- 2、本机作为服务端

使用远程调试的前提是服务器端和客户端的代码是一致的。

## 本机作为客户端

本机作客户端比较常用,需要在远端的服务器的 java 程序在启动时打开远程调试开关,

服务器端需要加上虚拟机参数 ■

作者: daimojingdeyu

本文更详细内容,请查看:

<http://developer.51cto.com/art/201102/245676.htm>

## ■ 编者按

我现在是自己做,但我此前有多年在从事软件开发工作,当回过头来想一想自己,觉得特别想对那些初学 JAVA/DOT .NET 技术的朋友说点心里话,希望你们能从我们的体会中,多少受点启发。

# 程序员们 不要想一辈子靠技术混饭吃

一、在中国你千万不要因为学习技术就可以换来稳定的生活和高的薪水待遇,你千万更不要认为哪些从事市场开发,跑腿的人,没有前途。

不知道你是不是知道,咱们中国有相当大的一部分软件公司,他们的软件开发团队都小的可怜,甚至只有 1-3 个人,连一个项目小组都算不上,而这样的团队却要承担一个软件公司所有的软件开发任务,在软件上线和开发的关键阶段需要团队的成员没日没夜的加班,还需要为测试出的 BUG 和不能按时提交的软件模块功能而心怀忐忑,有的时候如果你不幸加入现场开发的团队你则需要背井离乡告别你的女友,进行封闭开发,你平时除了编码之外就是吃饭和睡觉(有钱的公司甚至请个保姆为你做饭,以让你节省出更多的时间来投入到工作中,让你一直在那种累了就休息,不累就立即工作的状态)。更可怕的是,会让你接触的人际关系非常单一,除了有限的技术人员之外你几乎见不到做其他行业工作和职位的人,你的朋友圈子小且单一,甚至破坏你原有的爱情(想象一下,你在地做现场开发 2 个月以上,却从没跟女友见过一面的话,你的女友是不是会对你呲牙裂嘴)。

也许你拿到了所谓的白领的工资,但你却从此失去享受生活的自由,如果你想做技术人员尤其是开发人员,我想你很快就会理解,你多么想在

一个地方长期待一段时间,认识一些朋友,多一些生活时间的愿望。比之于我们的生活和人际关系及工作,那些从事售前和市场开发的朋友,却有比我们多的多的工作之外的时间,甚至他们工作的时间有的时候是和生活的时间是可以兼顾的,他们可以通过市场开发,认识各个行业的人士,可以认识各种各样的朋友,他们比我们坦率说更有发财和发展的机会,只要他们跟我们一样勤奋。(有一种勤奋的普通人,如果给他换个地方,他马上会成为是一个勤奋且出众的人。)

二、在学习技术的时候千万不要认为如果做到技术最强,就可以成为 100% 受尊重的人。

有一次一个人在面试项目经理的时候说了这么一段话:我只用最听话的人,按照我的要求做只要是听话就要,如果不听话不管他技术再好也不要。随后这个人得到了试用机会,如果没意外的话,他一定会是下一个项目经理的继任者。

朋友们你知道吗?不管你技术有多强,你也不可能自由的腾出时间象别人那样研究一下 LINUX 源码,甚至写一个 LINUX 样的杰作来表现你的才能。你需要做的就是按照要求写代码,写代码的含义就是都规定好,你按照规定写,你很快就会发现你昨天写的代码,跟今天写的代码有很多类似。本文未完,请阅读原文: ■

<http://developer.51cto.com/art/201111/299998.htm>

## ■ 编者按

51CTO 十月 Java 备忘录将记录本月有关 Java 的点点滴滴,本月 Java 业界比较重大的事件属于 Java 未来蓝图发布和 JavaFX 2.0 的发布。

# 10月Java备忘录：盘点Java主要成就

2011 年正在接近尾声, Oracle 公司也总结了 2011 年 Java 的主要成就。不光是在 Android 领域继续和 Google 展开赔偿诉讼,还在提升 Java 社区忠诚度和更新支持方面有了进步。

## 盘点 Java2011 年成就

2011 年甲骨文公司发布的 JDK 7, 可用于 Windows, Linux, Solaris 和嵌入式平台, 还发布了 Mac OS X 的开发人员预览版。从此 Java 迈进了 7 时代, 并预期未来的 Java 8 将会在云开发方面有长足的进步。JDK 8 将在 2013 年夏天发布, 会增加 Jigsaw, Lambda、JavaFX 3.0 等诸多功能。

在 2011 年 IBM, Apple, SAP, Azul 和 Twitter 加入了 OpenJDK。而 JRockit 团队和 HotSpot 团队已经成功被 Oracle 整合, 目前效率提升较大。

曾经一度销声匿迹的 JavaFX 也推出了 JavaFX 2.0, 提供了 Java 客户端用户界面 (UI) 技术的迁移路径。JavaFX 2.0 是先进的企业应用 Java UI 平台, 下一步 Java 将作为主要的富客户端平台进行发展。不过 51CTO 编辑很担心它在与 HTML 5、Flash 和 Silverlight 的竞争中, 能有多大的竞争力。

Java ME 也在慢慢发展, 试图跟上目前移动设备端的快速发展。甲骨文已建立了一个打通 Java 平台和 Java SE、Java ME 的沟通桥梁的战略计划, 同步发行在 Java SE 上运行的 Java ME API。此举是为实现嵌入式垂直市场的全覆盖。Java

平台将覆盖所有 CPU 类型。将致力于为垂直细分市场提供 API。

Oracle 还计划为 Java 推出一个公共云服务, 以完成其 PaaS 的战略目标。甲骨文 Java 云服务是一个为企业开发, 部署和管理 关键业务的 Java 平台, Java EE 应用平台将基于 Oracle WebLogic Server。它支持多个 Java 集成开发环境 (IDE), 包括甲骨文的 JDeveloper, NetBeans 和 Eclipse 的开发与部署。

Oracle 将在 Java 中更好支持 HTML 5, 以简化基于云的客户端 / 服务器互动应用程序开发。

## Java 未来发展蓝图发布

Oracle 在 JavaOne 大会上发布了 Java 未来的发展蓝图, 其中包括计划 2013 年推出的 Java 8。JavaScript 将会在未来得到改进, 以方便程序员进行开发。在下一代 Java 客户端中将会整合 JavaFX 3.0, 同时还将支持多点触控设备。

更多 Java 蓝图的细节, 请点击这里 >>

## Google 与 Java 专利权擦肩而过

2000 万美元能做什么? 估计这只是欧洲五大足球联赛中一个球星的价钱。而当年 Google 曾经有机会以 2000 万美元的价格, 从 Sun 手中买到 Java 为期三年的合作协议。倘若当年 Google 舍得花这 2000 万美元, 就没有现在 Oracle 什么事情了, 拉里也只能干着急。

可惜这没有发生, 导致 Google 不得不承诺自己



## 10 月 Java 备忘录: 盘点 Java 主要成就 II

在未来 Android 版本中将不会涉及侵犯 Java 专利的代码。猫鼠游戏还在继续, Google 和 Oracle 目前为止还没有达成协议。而不知道诺基亚推出 Windows Phone 手机后, Oracle 还能不能用 Android 要挟 Google。马上从 Google 身上割掉一块肉, 可能是 Oracle 最优的选择。

### Java 之父高斯林评价乔布斯“他是个疯子”

在天朝, 疯子就是疯子, 是要送进疯人院电击的。在美国, 疯子是个褒义词, 是偏执狂的代名词。世界上只有偏执狂才能成功, 所以“疯子”乔布斯成功了。Java 之父高斯林曾和乔布斯三次免谈, 每次都聊的很投机, 但是每一次聊完高斯林都告诉自己“不, 我不能为这个人工作, 他是个疯子!”。

其实高斯林也是个偏执狂, 这个老头当面呵斥 Oracle CEO 拉里, 并公开叫他“魔鬼拉里”。被 Oracle 炒掉后加入 Google, 但不久就跳槽了。看得出这个老头也是一个疯子, 一个执着于技术的偏执狂。

您对于 Java 在 2011 年 10 月的新动向有什么意见, 欢迎进入专用讨论贴讨论。地址:

<http://bbs.51cto.com/thread-891237-1.html>

### 2011 年 10 月 Java 新版工具推荐

JRuby 1.6.5, JRuby 是 Ruby 编程语言的 100% 纯 Java 实现, 它结合了 Ruby 简洁的语法以及 Java 丰富的类库。 下载地址

VisualVM 1.3.3, VisualVM 是一个集成多个 JDK 命令行工具的可视化工具。透过这个工具 Java 程序员可以更好的提升程序性能。 下载

Eclipse 4.2 (Juno) M3 继 Eclipse Foundation 发布了第一个支持 Java 7 的里程碑版 Eclipse 3.8M1 后, 近日 Eclipse 4.2 M3 也随之发布了! 下载

Oddjob 1.0, Oddjob 是一个采用 Java 语言开发的作业调度工具, 通过图形化界面或者是 XML 文件进行任务的设定, 支持按顺序执行、并行执行以及根据预设条件执行任务。 下载

本文其他部分, 请参考原文, 链接:

<http://developer.51cto.com/art/201110/299750.htm> ■

### 51CTO 精品杂志推荐《Linux 运维趋势》

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志, 内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。



本杂志长期处于探索期, 需要更多来自大家的意见与参与。如果您对这份电子杂志感兴趣, 那就请您下载阅读; 想要帮助我们做的更好, 请进入我们的 Linux 运维趋势技术圈留下您的宝贵意见建议。

读者讨论组: <http://g.51cto.com/linuxops/>

邮件订阅入口:

<http://os.51cto.com/art/201011/233915.htm>

投稿邮箱: [yangsai#51cto.com](mailto:yangsai#51cto.com)

发布周期: 每个月的第二个星期五

往期《Linux 运维趋势》下载汇总页:

<http://down.51cto.com/zt/71>

# Scala难在哪里？

我过去以为 Scala 很简单。它过去确实很简单,而且一直很简单,它是治疗 Java 里很多问题的良方。从“有些使用 Java 显的异常的困难或不可能的事,使用 Scala 却非常容易”的角度,Scala 是一种非常简单的语言。那么,Scala 难在哪里？

首先要说的是,我是一个 Scala 粉丝,我作为一个 Scala 语言的倡导者差不多有 5 年历史了。我写了不少 Scala 语言方面的书和文章。我曾在数十个公司里做过 Scala 和 Lift 框架项目的开发。我对很多的 Scala 项目进行过代码审查。

我过去以为 Scala 很简单。它过去确实很简单,而且一直很简单,它是治疗 Java 里很多问题的良方。从“有些使用 Java 显的异常的困难或不可能的事,使用 Scala 却非常容易”的角度,Scala 是一种非常简单的语言。Scala 处理集合问题超级的容易。业务逻辑的相互独立会使程序变得更容易维护,Scala 相对 Java 来说更方便达到这样的目标。

51CTO 推荐专题: Scala 编程语言

那么,Scala 难在哪里?下面是我能想出的最主要的几条:

◆ Scala 想要的东西太多。你可以拿 Scala 像 Java 那样编程。这是一种福气,也是一种诅咒,但我从长远的角度看,更多的是一种诅咒。关于它的面向对象 vs 面向函数的争议太多。对于小的开发团队,这些争议和你所采取的选择关系不大,但当你的团队有相当的人数,你试图教会这些 Java 程序员使用 Scala,而他们又非真心的想学时,这成了相当讨厌的事。Scala 语言的巨大优势会

在你使用函数式编程时不言自明的显露出来,但如果你只把自己当成面向对象的程序员,它的优势你是不可能看到的。对于这种情况,较少功能特征 / 可选性的语言 (例如 Java 或 Ruby) 就显得容易些。你不用费脑筋去做出选择。

◆ 集成开发工具对它的支持很弱,而且以后也不会改善。Scala 的 Eclipse 插件很差劲。从此我开始使用 Scala 语言五年来一直很差劲,它总是让人感觉“可以做的更好”,但却一直这样差劲。IntelliJ 对 Scala 的支持还凑合。但在 IDE 里需要使用各种模式的人会找不到一个好用的。Scala 的模式各式各样又互不关联,如果你不讨厌使用 Emacs 或 Vi 或 TextMate 编程,那使用 IntelliJ 开发 Scala 是个不错的选择。如果你期待着一个像 Java IDE 那样的东西,你找不到,而且永远找不到,因为 Scala 的强大能力是不能通过简单的模板表现出来的,你需要提供太多的信息资源给 IDE,它里面的类型安全 (TypeSafe) 检查的复杂,即使你银行里有 3 百万美元,也没有公司敢出来担保。

◆ Scala 的类型系统异常的强大,但它却让你茫然不知所措。在 ScalaDocs 里,类型符号复杂的让人恐怖。看着 `flatMap [B, That] (f: (A) Traversable[B])(implicit bf: CanBuildFrom[List[A], B, That]) : That`,是不是会让你有想逃的感觉? 这

## Scala 难在哪里？

是一个初学者每天都会用，一天用 20 次的方法，很恐怖吧。Scala 的文档须要一种调整来隐藏它的复杂度，让人们在实际使用中更容易的获取这 flatMap 的强大能力。类型系统以及相关的文档需要一种更简化的形式，把复杂性隐藏在程序包内，对最终用户要表现出简单的接口。（Adobe 架构师谈 Scala：功能强大但令人困惑）

◆ 当新程序员来维护老程序员写的 Scala 代码时，需要去理解代码中的风格和模式。Scala 的代码会使业务逻辑直接表现在最外层（而不是循环语句或复杂 IF 语句四处分布），如果代码中存在风格习惯，业务逻辑就不是那么直接。没有风格也是个问题，但最终，整个团队需要统一接受这样的风格模式。在 Ruby 和 Rails 编程中也是这样，hashmap 替代了所有其它种的编程方法。但在 Rails 里，风格是统一的（尽管没有类型检查），人们很容易理解，因为它就是这种“方式”。在 Java 里，代码模板由 IDE 生成，程序员养成了很容易发现其中的模式的能力。但在 Scala 中却不是这样，各种风格迥异，每个开发团队里都不相同。

我知道有很多的开发团队，在他们的团队组织形式里，采用 Scala 语言会比使用 Java 或 Ruby 或其它语言要合适的多，Twitter 公司就是这样的典型例子。他们需要一个简洁的，具有类型检查的，高性能的语言和运行环境。Scala 满足了他们的这些需求。Foursquare 公司以 Scala 的难度作为一种过滤制度。你只有学好了 Scala 语言才能在这个公司立足。

但如果你的团队的技术水平很一般，Scala 也许对你们公司来说并不是一个好的选项。Scala 的难度导致很陡的学习曲线，会遭到原有的程序

员的反对，形成不了统一的风格。你需要一个强有力的 CTO 或架构师来强迫这种风格，而不是让他们自己从书中学习。

那么，如何能看出 Scala 在你们的团队中会是很“简单”还是很“难”呢？

◆ 如果你的公司在 JavaOne 大会，或 OSCON，Strangle Loop，或 QCon 大会上出席发言的人：Scala 对于你们来说会很简单

◆ 如果吃饭时间你们还在讨论如何从一个普通程序员成长成高级程序员：Scala 对你们来说会很难

◆ 如果需要，你可以用 NotePad 编程：容易

◆ 当看到“Zed Shaw”时，你的程序员面无表情或连说 3 声“万福玛利亚！”：Scala == 难

◆ 程序员在 Twitter 上关注 Dean Wampler：Scala 简单

◆ 你的程序员 9:15 到公司，晚上不看有没有邮件：难

现在你们知道了。我完全同意这样的观点：对于水平一般的团队，Scala 很难。并不是它本身很难，而是因为它在水平一般的团队中不会产生那种由技术很好的人组成的团队中产生的短期或长期的益处。

一些评论：

◆ 不错，Scala 的类型系统很强大，由它产生了很多优美的程序代码，例如 Scala 的集合。。但是，对于 Scala，从一个语言设计者 / 程序库创造者的角度，和从一个普通程序员的角度，他们的需求是不同的。

本文未完，阅读全文请链接：

<http://developer.51cto.com/art/201111/300820.htm> ■

# 项目经理不可不知的“三边六拍”

作者 / 张传波

“三边”是指：边计划、边实施、边修改，“六拍”是指：拍脑袋、拍肩膀、拍胸口、拍桌子、拍屁股、拍大腿！“三边六拍”很形象，引人深思！

“三边六拍”并非本人原创，最开始我是从我的领导那听到的，我觉得很有意思，很形象，很贴实际。下面介绍一下什么是项目管理“六拍”，以及我的一些感受；然后介绍“三边”和我的体会。

## 第一拍：拍脑袋

客户领导与软件公司领导在一起，“拍脑袋”确定了一个项目，于是项目马上就要开始了，你的麻烦马上就要来了。

“拍脑袋”主要说明的问题是没有想清楚为什么要做这个项目，不怎样考虑可行性和最终效果，就冲动地让项目上马了。

我的感受：

作为项目经理的你，往往无法干预这个“拍脑袋”的过程，你想你的领导下拍脑袋能拍好一点，恐怕也不太可能。你能做到的事情就是，搞清楚项目的背景、搞清楚你的领导和客户的领导做这个项目的动机是什么？

项目要成功，你必须同时满足你的领导和客户的想法。当然双方领导的想法可能都不确定，这可能会让你很被动，但也给你带来机会来“控制”这些事情，前期虽然是“拍脑袋”安排这个项目，但你可以在后期充分与双方领导沟通，逐步将领导脑袋中不明确的东西明确（当然难度超高），让项目有成功的机会。

## 第二拍：拍肩膀

领导安排项目工作给你时，往往是语重心长地说类似这样的话：这个项目对公司如何如何重要，你是公司中非常重要的人物，这个项目没有你不能成功……拍你的肩膀，就是和你套亲切和信任，有时候还会请你甚至是项目组吃饭，表示对大家的感谢和信任。

我的感想：

咱们做软件开发的，其实不是傻的，这些门面功夫其实让我觉得很恶心，而你遇到这样的情况时，你又必须虚伪地做下面的第三拍的事情，更加恶心！但如果我是领导，我会如何安排项目工作下去呢？会不会也是这样做这些恶心的门面功夫？

## 第三拍：拍胸口

有些朋友，可能真的是以为领导很重视你，义无反顾的“拍胸口”保证没有问题！

有些朋友就比较聪明，他知道这个项目你是不可能推脱的（除非你想走人），但他又知道任务艰巨，他会在拍胸口的同时，要求增加人手和延长工期，往往会得到老板的一些妥协。

而我是属于比较蠢的一种，我会直接跟领导说这个项目的一些重大问题，有些项目我了解到项目“拍脑袋”过程后，我甚至会说这个项目不可能成功。我这样做往往惹领导不高兴，觉得我不听话，认为我对本项目没有信心等等。

我的感想：

做人真难，我不是随便承诺的人，我能列出



## 项目经理不可不知的“三边六拍” II

项目的具体问题,说明已经认真思考了,有问题自然有努力的方向,没有问题其实问题更大,不知道领导是怎样想的?拍胸口真的是很痛苦的事情啊!

### 第四拍:拍桌子

项目进展不理想,领导拍桌子骂人!

领导拍桌子骂人,可能是真的很火,忍不住拍了,也可能是要给你一种威慑感,让你更有压力的干活。

也有的领导很厉害,会忍住怒火,他知道目前只能靠你,等项目完了后才“收拾”你!

我的感想:

用工厂模式来管理软件项目,其实是很傻的做法。软件项目是充满创造力的事情,无聊的高压,往往让项目组不思考如何更简单,更少工作量,而疲于奔命,不断返工。而老板见到你拼命加班,虽然工作成功不尽理想,但至少老板心里会舒坦一些。这是多么无聊的怪现象啊!

如果遇到厉害的领导,暂时忍住怒火,继续对你“以礼相待”。有人会趁机“要挟”一把,这时候要求加薪,90%是没有问题的,但你等着项目后被开掉吧,而且可能会被开得很惨。

### 第五拍:拍屁股

现在不是奴隶社会,我又不是终身卖给你,老子不打东家可以打西家,你可以发飙,老子何尝不可以!老子走人,你自己管你这破项目吧!

我的感想:

不管东家还是西家,其实家家都是一个鬼样,你只是从一个火坑跳到另外一个火坑而已。不要轻易拍屁股,要思考自己是否值得留下来!

凡事没有完美,出来打工更加不可能事事顺意,从你的职业规划思考,你值不值得继续为这个项目奋斗?当然要记住,做人要有基本的职业道德,他不仁,我不会不义!

### 第六拍:拍大腿

领导将项目经理气走了,项目无人打理,于是拍大腿后悔。

我的感想:

早知如此,何必当初呢?但很多领导还是不断地重复着“六拍”,领导还真的是不好当啊!

**“三边”是指:边计划、边实施、边修改。**

很多项目往往没有通盘规划就仓促上马,紧张的项目工期,太多的不清楚因素,项目组在高压下,往往是见一步走一步,一边做一边修改。最终项目效果也不会很好,不但项目组痛苦,客户的利益也会受到伤害,最后结果就是“双输”了!

实际上很多项目可能逃不过“三边”命运,因为项目上马时,需求是不确定的,技术往往也是不确定的,但工期是限死的,项目的预算也很可能是限死的,在“两大限死、两不确定”的框架下,“三边”难以避免。我做过的项目,几乎都不可能从一开始就能得到全面准确的需求,都不可能在项目初期就可以制定长期而周长的计划,小版本迭代是我惯常的做法。其实敏捷就有具备“三边”的特点,不过不是“见一步走一步”,而是有一个长远的战略性规划,通过一个个小版本逐步接近我们的目标,保持灵巧的同时也能预测变化。

所以我对“三边”的感想是:“三边”不代表低水平,关键你是敏捷地“三边”还是被动地“三边”? ■

# 10月Web技术最前沿:JavaFX 2发布

作者 / 七武海

在进入主题之前让我们先一起缅怀本月离我们而去的业界牛人: 苹果公司的创始人史蒂夫·乔布斯、C语言之父同时也是 Unix 之父丹尼斯·里奇、手机之父马蒂·库珀、Lisp 语言发明者人工智能之父 John McCarthy。51CTO 在第一时间发布他们离去的消息后, 51CTO 网友也都以各种方式表达了对这些牛人的尊敬和缅怀。

下面是本月 web 技术最前沿:

## jQuery 1.7 RC1 发布

51CTO 推荐专题: jQuery 从入门到精通

jQuery 1.7 RC1 在 2011 年 10 月 24 日已经发布了, 这次大幅更新将给大家带来例如 HTML 5 解决方案等好用的新功能。大家可以在自己的应用中测试这些代码, 以使得能在最终发布之前能处理掉这些主要问题。测试请不要过于简单, 你可以在 jQuery CDN 中拿到这些测试代码:

<http://code.jquery.com/jquery-1.7rc1.js>

<http://code.jquery.com/jquery-1.7rc1.min.js>

请把这些代码放到您已经存在的应用中。如果您从 jQuery 1.7 RC1 中发现 BUG 请及时通知我们以便于修复问题。

jQuery 1.7 RC1 英文版本的详细更新日志, 请点击: <http://blog.jquery.com/2011/10/24/jquery-1-7-rc1-released/>

更多详细请点击: <http://developer.51cto.com/art/201110/298940.htm>

## JavaFX 2.0 正式版发布

JavaFX 2.0 是 JavaFX 一个主要的升级版本,

为下一代的 UI 平台而设计, 提供强大的基于 Java 的 UI 能力, 可处理大规模数据驱动的商业逻辑。

JavaFX 2.0 包含非常丰富的 UI 控件、图形和多媒体特性用于简化可视化应用的开发, 新增的 WebView 可直接在应用中嵌入网页; 另外 2.0 版本允许使用 FXML 进行 UI 定义, 这是一个脚本化基于 XML 的标识语言。

更多关于 JavaFX 2.0 的消息请看官方发行说明 [http://blogs.oracle.com/thejavatutorials/entry/javafx\\_2\\_0\\_released](http://blogs.oracle.com/thejavatutorials/entry/javafx_2_0_released)

## Ruby 的文档系统 RDoc 3.10 发布

RDoc 3.10 发布了, RDoc 是针对 Ruby 源代码的文档系统。跟 Javadoc 一样, RDoc 可以读取原文件, 分析源代码中的语法信息和注释文本, 生成 HTML 格式的文档。

另外, 该版本还修复了一些 Bug, 详细信息参阅: RDoc 发行说明

Rdoc 文档: <http://rdoc.rubyforge.org/>

下载地址: <https://github.com/rdoc/rdoc>

## Perl 的 Web 框架 Mojolicious 2.0 发布

Mojolicious 是下一代的 Perl 语言的 Web 编程框架, 提供强大的功能而无需第三方包依赖, 功能包括: RESTful 的路由、插件管理、Perl 风格的模板系统、会话管理、签名的 cookie、静态文件服务、测试框架、文件级的 Unicode 支持等等。

Changes:

1. Perl 5.10.1: 提升了性能
2. Morbo: 可自启动的开发用 Web 服务器

## 10 月 Web 技术最前沿 :JavaFX 2 发布 II

3. 内置对 libev 事件循环的支持
4. 使用事件对框架进行扩展
5. Mount 插件允许嵌入整个应用
6. 产品级的 Web 服务器更加友好
7. Collections make Web scraping fun.
8. 可方便进行 WebSocket 编程测试

下载请点击这里

<http://www.oschina.net/p/mojolicious>

### CakePHP 2.0 发布

CakePHP 开发团队已经发布了 CakePHP 2.x 系列的一个版本 CakePHP 2.0。在该版本中不再支持 PHP4, CakePHP 团队对其代码库进行了重构,以便严格遵循 PHP 5 规范 (包括 5.2 及以后版本)。

CakePHP 2.0 的新亮点便是支持 PHP 5 特性,包括 PHP 5 的异常、标准库、JSON 编码 (json\_encode) 及新引入的 PDO(PHP Data Objects) 数据访问接口。PSR-0 命名约定已经被采纳,所有的类都可被映射至一个与类名相同的文件。对象现在已可作为 CakePHP 库注入,正如开发人员所说的“没有借口再去修改核心文件了”。开发团队同时表示,由于 lazy-loading(延迟加载)的广泛使用, CakePHP 现在要比以前快得多,甚至在调试模式下也如此。

CakePHP 2.0.0 与 1.3 分支的 CakePHP 1.3.13 是同时发布的。1.3.13 是对 1.3.12 的一个小更新,主要修复了其中引入的两个 regression 问题。CakePHP 1.3.13 还移除了 EmailComponet 组件中的 MIME 类型通知,以解决投递问题,同时改进 SecurityComponent 和 TranslateBehavior 组件。更多信息参见 CakePHP 2.0 和 1.3.13 的改进日志。

CakePHP 2.0 Changelogs:

<http://cakephp.org/changelogs/2.0.0>

CakePHP 1.3.13 Changelogs:

<http://cakephp.org/changelogs/1.3.13>

下载地址:

CakePHP 2.0:

<https://github.com/cakephp/cakephp/zipball/2.0.0>

CakePHP 1.3.13:

<https://github.com/cakephp/cakephp/zipball/1.3.13>

### Nginx 1.1.6 开发版发布

Nginx 1.1.6 开发版发布了。该版本的改进和 Bug 修复包括:

改进:

◆ 内部 API 的变化:现在模块的上下文数据将被清除,同时重定向到指定位置。

◆ 改进:如果服务器的 upstream 失败,将在 fail\_timeout 后只给它发送一个请求,如果服务器成功地响应请求,则该服务器可被认为是活动的。

◆ 改进:现在已避免 0x7F-0x1F 等字符,如 access\_log 中的 \xXX 字符。

◆ 特性:“proxy/fastcgi/scgi/uwsgi\_ignore\_headers”指令支持以下附加值: X-Accel-Limit-Rate、X-Accel-Buffering 和 X-Accel-Charset。

◆ 特性:减少了使用 SSL 时的内存消耗。

Bug 修复:

◆ 修复一些 UTF-8 字符不正确处理问题。

◆ 修复了如果没有匹配的位置被定义,则 server 级别的 ngx\_http\_rewrite\_module 指令将会执行两次的问题。

下载: <http://nginx.org/en/download.html> ■

# Node.js提速指南

没有人奢望基于 Node.js 的各类系统能够一统 web 服务器这一庞大领域,但 Node 自身所具备的灵活性确实使其身影屡屡出现在各种各样的任务处理流程之中。那么到底是哪些特色使得 Node 从以往那些 web 框架与平台中脱颖而出呢?



Node.js 又被简称为 Node,作为一款针对 web 开发者推出的 web 应用程序平台,它已经在过去的一年中得到了相当令人满意的普及度。没有人奢望基于 Node.js 的各类系统能够一统 web 服务器这一庞大领域,但 Node 自身所具备的灵活性确实使其身影屡屡出现在各种各样的任务处理流程之中。那么到底是哪些特色使得 Node 从以往那些 web 框架与平台中脱颖而出呢?归纳起来有两点,基于事件以及 JavaScript。

51CTO 推荐专题:Node.js 专区

## 基于事件

### 传统的阻断程序

传统的 web 请求处理流程分为几步,即接收请求、进行解析、等待必要资源载入、处理(无论需要多长时间)以及返回响应。由于整个过程中充斥着大量等待环节,因此在同时处理两个或者两个以上请求时必须为每个请求分配一个独立的线程来满足执行需求。需要处理的请求越多,我们需要的线程就越多,同时我们还要为每个线程的管理投入大量额外的资源。

基于事件的框架则采取完全不同的解决方案,不过这类框架同时也要求我们使用不同的编码方式。它们所利用的正是许多服务器应用程序浪费在等待 I/O 上的时间,尝试将这些时间用在实实在在的工作上。执行线程实际上只

有一个,但程序员将自己的代码有机地加以分解,并把每一块视为一次“事件”的出现。举例来说,打开一个文件当然会带来 I/O 时间,因此在以事件为主导的系统中,我们会下达“请开始打开一个文件,当文件打开工作完成后,再调回该功能。”这样框架就会着手打开文件,并把过程中需要用到的功能记录下来,最后等待操作系统发来的文件打开完成通知。一旦通知送达,该事件将立即被触发,转而调用所需功能。

```
open_file(filename, open_done_function)
open_done_function() { write(open_file, data, write_done_function) }
close_file_function() { close(open_file) }
```

某种风格的非阻断事件驱动虚拟代码

现在大家的第一反应可能是:“这难道不会使我的代码在互连功能方面变得乱七八糟吗?”答案是肯定的,如果大家的代码在表现力上有所不足,那么这种负面现象的确会发生。举例来说,如果各位选择的是能够处理匿名函数的语言,那么代码可能会变成如下所示:

```
open_file(filename, open_done_function)
open_done_function() { write(open_file, data, write_done_function) }
close_file_function() { close(open_file) }
```

另一种风格的非阻断事件驱动虚拟代码

真正的区别在于,执行时间在操作完成时并不会立即中止,而是会在底层操作系统通知任务结束后记录接下来所要进行的任务。事件驱动类编程能够通过一系列语言实现,可读性与易用性



# 为什么我如此迷恋Lisp语言？

本文是外刊 IT 评论从 Why I love Lisp 这篇文章翻译而来。作者认为只有当你独自一人深入荒漠,用树枝在黄沙上为自己喜欢的 Lisp 方言写解释器时,你才成为一名真正的 Lisp 程序员。

这篇文章是我在 Simplificator——我工作的地方——的一次座谈内容的摘录,座谈的题目叫做“为什么我喜欢 Smalltalk 语言和 Lisp 语言”。在此之前,我曾发布过一篇叫做“为什么我喜欢 Smalltalk?”的文章。



大漠黄沙 by Guilherme J ó fili

Lisp 是一种很老的语言。非常的老。Lisp 有很多变种,但如今已没有一种语言叫 Lisp 的了。事实上,有多少 Lisp 程序员,就有多少种 Lisp。这是因为,只有当你独自一人深入荒漠,用树枝在黄沙上为自己喜欢的 Lisp 方言写解释器时,你才成为一名真正的 Lisp 程序员。

目前主要有两种 Lisp 语言分支: Common Lisp 和 Scheme,每一种都有无数种的语言实现。各种 Common Lisp 实现都大同小异,而各种 Scheme 实现表现各异,有些看起来非常的不同,但它们的基本规则都相同。这两种语言都非常有

趣,但我却没有在实际工作中用过其中的任何一种。这两种语言中分别在不同的方面让我苦恼,在所有的 Lisp 方言中,我最喜欢的是 Clojure 语言。我不想在这个问题上做更多的讨论,这是个人喜好,说起来很麻烦。

Clojure,就像其它种的 Lisp 语言一样,有一个 REPL(Read Eval Print Loop) 环境,你可以在里面写代码,而且能马上得到运行结果。例如:

```
5
;=> 5
"Hello world"
;=> "Hello world"
```

通常,你会看到一个提示符,就像 user>,但在本文中,我使用的是更实用的显示风格。这篇文章中的任何 REPL 代码你都可以直接拷贝到 Try Clojure 运行。

我们可以像这样调用一个函数:

```
(println "Hello World")
; Hello World
;=> nil
```

程序打印出“Hello World”,并返回 nil。我知道,这里的括弧看起来好像放错了地方,但这是有原因的,你会发现,他跟 Java 风格的代码没有多少不同:

## 为什么我如此迷恋 Lisp 语言？ II

```
println("Hello World")
```

这种 Clojure 在执行任何操作时都要用到括弧：

```
(+ 1 2)
;=> 3
```

在 Clojure 中,我们同样能使用向量 (vector):

```
[1 2 3 4]
;=> [1 2 3 4]
```

还有符号 (symbol):

```
'symbol
;=> symbol
```

这里要用引号 (), 因为 Symbol 跟变量一样, 如果不用引号前缀, Clojure 会把它变成它的值。list 数据类型也一样:

```
'(li st)
;=> (li st)
```

以及嵌套的 list:

```
'(l (i s) t)
;=> (l (i s) t)
```

定义变量和使用变量的方法像这样:

```
(def hello-world "Hello world")
;=> #'user/hello-world

hello-world
;=> "Hello world"
```

我的讲解会很快,很多细节问题都会忽略掉,有些我讲的东西可能完全是错误的。请原谅,我尽力做到最好。

在 Clojure 中,创建函数的方法是这样:

```
(fn [n] (* n 2))
;=> #<user$eval1$fn__2 user$eval1$fn__2
@175bc6c8>
```

这显示的又长又难看的東西是被编译后的函数被打印出的样子。不要担心,你不会经常看

到它们。这是个函数,使用 `fn` 操作符创建,有一个参数 `n`。这个参数和 2 相乘,并当作结果返回。Clojure 和其它所有的 Lisp 语言一样,函数的最后一个表达式产生的值会被当作返回值返回。

如果你查看一个函数如何被调用:

```
(println "Hello World")
```

你会发现它的形式是,括弧,函数,参数,反括弧。或者用另一种方式描述,这是一个列表序列,序列的第一位是操作符,其余的都是参数。

让我们来调用这个函数:

```
((fn [n] (* n 2)) 10)
;=> 20
```

我在这里所做的是定义了一个匿名函数,并立即应用它。让我们来给这个函数起个名字:

```
(def twice (fn [n] (* n 2)))
;=> #'user/twice
```

现在我们通过这个名字来使用它:

```
(twice 32)
;=> 64
```

正像你看到的,函数就像其它数据一样被存放到了变量里。因为有些操作会反复使用,我们可以使用简化写法:

```
(defn twice [n] (* 2 n))
;=> #'user/twice

(twice 32)
;=> 64
```

我们使用 `if` 来给这个函数设定一个最大值:

```
(defn twice [n] (if (> n 50) 100 (* n 2)))
```

`if` 操作符有三个参数:断言,当断言是 `true` 时将要执行的语句,当断言是 `false` 时将要执行的语句 (未完) 查看详细:

<http://developer.51cto.com/art/201110/297020.htm>

## ■ 编者按

作为 140 个字的缔造者, twitter 太简单了,又太复杂了,今天就结合网络上的一些资料,来浅谈一下我对 twitter 网站架构的学习体会,希望给路过的朋友一点启示。—— 一个寂寞的分享者

# 揭秘Twitter网站架构

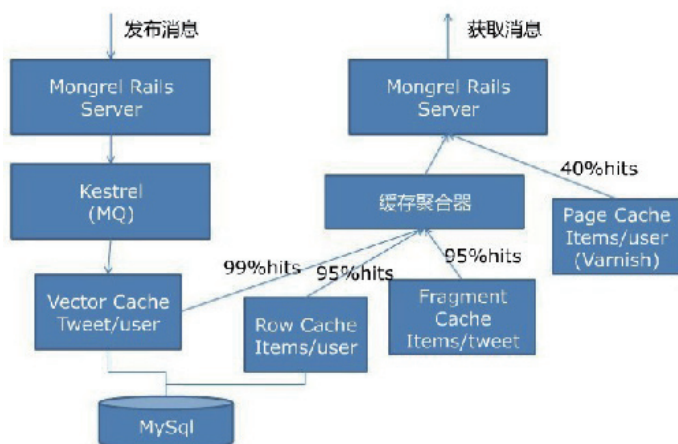
## 一、twitter 网站基本情况概览

◆ 截至 2011 年 4 月, twitter 的注册用户约为 1.75 亿,并以每天 300000 的新用户注册数增长,但是其真正的活跃用户远远小于这个数目,大部分注册用户都是没有关注者或没有关注别人的,这也是与 facebook 的 6 亿活跃用户不能相提并论的。

◆ twitter 每月有 180 万独立访问用户数,并且 75% 的流量来自 twitter.com 以外的网站。每天通过 API 有 30 亿次请求,每天平均产生 5500 次 tweet,37% 活跃用户为手机用户,约 60% 的 tweet 来自第三方的应用。

◆ 平台: Ruby on Rails 、 Erlang 、 MySQL 、 Mongrel 、 Munin 、 Nagios 、 Google Analytics 、 AWStats 、 Memcached

下图是 twitter 的整体架构设计图:



## 二、twitter 的平台

twitter 平台大致由 twitter.com、手机以及第三方应用构成,如下图所示:



其中流量主要以手机和第三方为主要来源。

◆ Ruby on Rails: web 应用程序的框架  
◆ Erlang: 通用的面向并发的编程语言,开源项目地址: <http://www.erlang.org/>

◆ AWStats: 实时日志分析系统: 开源项目地址: <http://awstats.sourceforge.net/>

◆ Memcached: 分布式内存缓存组建  
◆ Starling: Ruby 开发的轻量级消息队列  
◆ Varnish: 高性能开源 HTTP 加速器  
◆ Kestrel: scala 编写的消息中间件,开源项目地址: <http://github.com/robey/kestrel>

◆ Comet Server: Comet 是一种 ajax 长连接技术,利用 Comet 可以实现服务器主动向 web 浏览器推送数据,从而避免客户端的轮询带来的性能损失。

## 揭秘 Twitter 网站架构 II

- ◆ libmemcached: 一个 memcached 客户端
- ◆ 使用 mysql 数据库服务器
- ◆ Mongrel: Ruby 的 http 服务器, 专门应用于 rails, 开源项目地址:

<http://rubyforge.org/projects/mongrel/>

- ◆ Munin: 服务端监控程序, 项目地址:

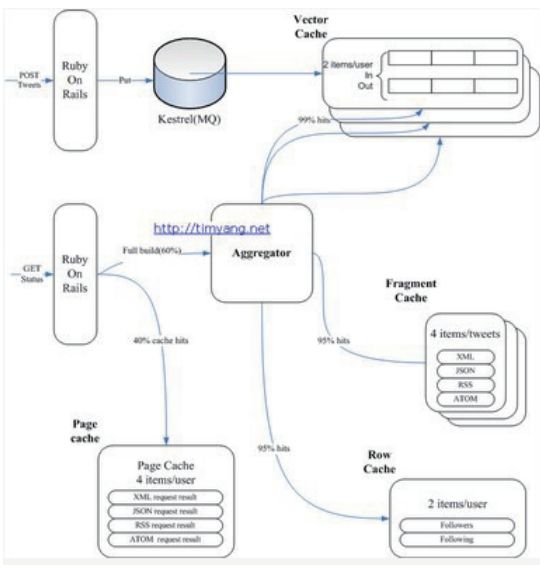
<http://munin-monitoring.org/>

- ◆ Nagios: 网络监控系统, 项目地址:

<http://www.nagios.org/>

### 三、缓存

讲着讲着就又说到了缓存了, 确实, 缓存在大型 web 项目中起到了举足轻重的作用, 毕竟数据越靠近 CPU 存取速度越快。下图是 twitter 的缓存架构图:



大量使用 memcached 作缓存

- ◆ 例如, 如果获得一个 count 非常慢, 你可以将 count 在 1 毫秒内扔入 memcached
- ◆ 获取朋友的状态是很复杂的, 这有安全等其他问题, 所以朋友的状态更新后扔在缓存里面

不是做一个查询。不会接触到数据库

- ◆ ActiveRecord 对象很大所以没有被缓存。Twitter 将 critical 的属性存储在一个哈希里并且当访问时迟加载

- ◆ 90% 的请求为 API 请求。所以在前端不做任何 page 和 fragment 缓存。页面非常时间敏感所以效率不高, 但 Twitter 缓存了 API 请求

在 memcached 缓存策略中, 又有所改进, 如下所述:

- 1、创建一个直写式向量缓存 Vector Cache, 包含了一个 tweet ID 的数组, tweet ID 是序列化的 64 位整数, 命中率是 99%

- 2、加入一个直写式行缓存 Row Cache, 它包含了数据库记录: 用户和 tweets。这一缓存有着 95% 的命中率。

- 3、引入了一个直读式的碎片缓存 Fragment Cache, 它包含了通过 API 客户端访问到的 tweets 序列化版本, 这些 tweets 可以被打包成 json、xml 或者 Atom 格式, 同样也有着 95% 的命中率。

- 4、为页面缓存创建一个单独的缓存池 Page Cache。该页面缓存池使用了一个分代的键模式, 而不是直接的实效。

### 四、消息队列

- ◆ 大量使用消息。生产者生产消息并放入队列, 然后分发给消费者。Twitter 主要的功能是作为不同形式 (SMS, Web, IM 等等) 之间的消息桥 (未完)

查看详细:

<http://developer.51cto.com/art/201110/296071.htm>



# 死诸葛吓走生仲达？移动之战刚开打

《三国演义》里有段经典的典故：死诸葛吓走生仲达。讲的是，蜀军主帅诸葛亮病死军中，蜀将姜维等遵照诸葛亮遗嘱，秘不发丧，缓缓退军。魏军主帅司马懿（字仲达）率军追击，见蜀军帅旗飘扬，孔明羽扇纶巾坐在车里。司马懿怀疑是孔明用计诱敌，赶紧策马收兵……

景琦



图片来自互联网

在移动互联网领域称得上诸葛亮的，最契合的就是乔布斯了。相比三国时期的通讯水准，乔教主的死讯真伪自然不会成为各竞争对手猜疑的环节，但是和三国时期相比，移动互联网的用兵却如出一辙。

为避乔布斯去世的锋芒，也有称是为悼念老对手，竞争对手们推迟了新品发布时间，但毫无疑问，智能手机之战并不会因悼念乔布斯而停止，休战只是缓兵之计。随着媒体报道的热度退却，“司马懿”们终于亮剑，搭载谷歌新 Android 4.0 操作系统的手机发布，诺基亚携手微软也推出了酝酿已久的 WP7 手机。媒体一顿造势之后，令人感慨后起之秀的反扑之猛。

## 教主英魂尚在

诸葛亮被罗贯中神化了不少，但是只是神化，毕竟人不是神，人会死。乔布斯犹如诸葛孔明，

一生运筹帷幄、精彩不断，俨然创造了苹果教，但教主是人不是神，教主也会死。留下的故事被传颂着，留下的战事还要继续打。

苹果给人们的惊喜多是超出你的预估，但本月 5 日，苹果 iPhone 4S 的发布令果粉和投资者们真的失望不少。6 日，乔布斯病逝，苹果未来的不确定性陡增。但笔者认为，至少从现在的市场反响来看，死诸葛还是能吓走生仲达的，用的计策就是 Siri——这个 iPhone 4S 的虚拟助手。

我们最初忽略了 Siri 的功能，或许仅把 Siri 语音服务跟语音识别弄混了。事实上，语音识别只是将用户的声音翻译成了文本，随着时间的推移，苹果的 Apple TV 或智能电视也将采用 Siri 语音识别功能。届时，人们将拥有一台可以“说话”的电视，这种乐趣是大多数人类还不曾拥有的，这种感觉太难想象了。

Siri 是个乔教主留给我们的礼物，也是令“司马懿”们胆寒的“妙计”。人机对话已不仅仅存在于好莱坞的电影里，人工智能真的走进了我们的生活，调戏它或将成为一种时尚（推荐阅读：苹果 Siri 的 10 种“正经”用法），虽然它还有待完善。但给人们的感觉是，亲眼看见了科技的进步。

## 死诸葛吓走生仲达？移动之战刚开打 II



图片来自互联网

## 谁将笑到最后？

“司马懿”们不傻，当年的他其实是腹背受敌。明着，要抵挡诸葛亮的六出祁山、光复汉室；暗着，要防着曹魏后裔们对司马家族的猜疑。笔者猜想，他不一定敢在不确定的时机内全歼蜀军，功高盖主的功臣一般下场都很惨。被诸葛亮玩了个遍的司马懿是否在装疯卖傻？不知道，反正人家笑到了最后。

乔布斯死前，移动互联网市场及话题都是围着他转的，只要他活着，苹果产品就不愁市场。对手的产品真的比他家差么，不好说，但在乔帮主的光环下，每个人都抬不起头来。有的已经被窝死了，比如 webOS。光环退却后的苹果还是很有实力的，但前途未卜。“老诸葛”死了，“司马懿”们似乎等到了时机。

谷歌，收购摩托罗拉移动就是其平复通讯行业内乱的标志，专利就像硬伤一样鞭笞着谷歌，收购后的他终于可以全力对付这个老对手了。新 Android 4.0 的光芒虽然闪耀，但其理念尚不足以撼动苹果的 Siri 优势，就第一仗来说，还是乔教主的遗作占了上风。但路还很长，安内后的谷歌已开始攘外，Android 4.0、NFC、脸部识别、多平台整合……当然，这一切，仅仅是个开始。

诺基亚，曾经的老大，曾经软硬兼备的顶级

移动厂商，曾经不可一世的王者，被当今的移动互联网打的溃不成军。现已联手微软的诺基亚，已放弃 meego、塞班（智能手机领域），并且在近日发布 WP7 手机。凭借曾经良好的口碑，我们有理由对其未来报以美好的憧憬，不少投资人和消费者都有此同感。不少开发团队也钟爱这个老牌王者，但他们需要看到诺基亚新手机有实质性的改变，才能重拾信心……当然，这一切，仅仅是个开始。

索尼，刚刚以 14.7 亿美元收购了索尼爱立信 50% 的股权，使该手机制造商成为索尼的全资子公司。在智能手机尚未普及的时代，索爱也算是手机领域的一个国际时尚品牌。但随着苹果的崛起，各大老牌厂商都被打了个措手不及，有的走投无路投靠他人（如摩托罗拉移动），有的早已销声匿迹。索尼作为数码、家电的龙头老大，理应对这个有着自己血统的品牌负责，至少不能让其顶着自己半个光环而为自己的品牌抹黑。收购来的很及时，希望索尼能像做消费电子一样的水准来做个新手机……当然，这一切，仅仅是个开始。

除了以上的几家变化很大的公司，三星、HTC、黑莓等都不能说没有翻盘的机会，大家都在努力，教主死后满眼都是机会。

结语：“司马懿”们终于等到了“老诸葛”的死亡，虽然大家都还没有能拿得出手对付老教主 Siri 的兵力，但他们已开始蠢蠢欲动。蜀国的势力还有多大，是否能有个姜维（库克）继承武侯（乔布斯）遗志，战局结果如何？这都是个未知数。《三国》的结局是，诸葛死后蜀国不久战败，司马家族笑到了最后。但这可不是罗贯中笔下的《三国》，鹿死谁手？且听下回分解。

原文链接：<http://mobile.51cto.com/hot-299385.htm>

# 美女与极客之战:Mango VS Android

核子可乐译

新款微软 Windows Phone 只能算作一轮小小的升级,既缺乏企业级安全性保障又没有丰富的应用程序支持;相比之下 Android 智能手机系统则算得上更明智的选择。但是依旧有很多人认为 Windows Phone Mango 非常优秀,称之为“美女”一般的操作系统,而最新的 Android 2.3 不过是一个“极客”操作系统而已。



(美女与极客)来自互联网

## Windows Phone Mango 对 Android 2.3: 定位支持

Windows Phone 与 Android 系统都支持 GPS 定位,并且都能够以基于 Wi-Fi 信号进行三角定位。它们还具备同等效果的地图应用程序,能够帮助我们找到当前目的地、提供方向指引并以各种方式为我们导航。

Android 系统中的导航应用程序提供了 Windows Phone(包括 iOS)所无法比拟的功能性

补充,要达到同样的效果大家恐怕要额外购买单独的导航应用才行。

Windows Phone 与 Android 系统都允许开发人员在应用程序中添加定位信息,因此定位可以看作是另一项原生功能。这两款操作系统也都允许大家对位置隐私加以控制,但控制水准并不理想:在整台设备上禁用或启用 GPS 及 Wi-Fi 定位服务。Windows Phone 及 Android 应用程序会询问是否被允许使用用户当前所处的位置,但是与 iOS 不同,其中并没有一套核心机制来管理这些位置的使用权限。

**优胜者:** Android,这次胜利的得来要归功于其自有导航应用程序。如果不考虑这一点,两者基本上旗鼓相当。

## Windows Phone Mango VS Android 2.3: 网页与互联网

谷歌有 HTML 5 技术的强力支持,因此其提供的网页浏览器在功能方面优势明显。与谷歌相比,微软支持新标准的脚步迟缓了许多。虽然根据测试网站的数据, Mango 与竞争对手相比,HTML 5 兼容性已经取得重大突破,但它仍然落后于其它移动浏览器。其实 Windows Phone 与 Android 浏览器有本质不同的 UI。它们的接口都相当少,URL 地址栏也得以保留;在 Android 中可以通过一个图标按钮打开书签页面,而在 Windows Phone 上则可以通过一个图标按钮刷新网页。在实现返回浏览历史记录方面所使用的都是实体后退按钮。



## 美女与极客之战 :Mango VS Android II

在 Windows Phone 与 Android 中,我们可以直接选定网页上的文字及图形,但只有 Windows Phone 允许大家保存当前选定的内容。Windows Phone 与 Android 浏览器都提供了控制 cookie 及访问历史的设定,但 Mango 不具备用来管理其它个人信息,诸如缓存、表彰数据、密码等功能的选项。当我们不想访问网站的移动设备优化版本时,Windows Phone 会告知网站发起访问的设备是台式机而非移动设备。

尽管 Android 2.3 系统中没有预装,但 Adobe Flash Player 仍然可以在 Android Market 中免费下载获得。我发现最新的 Flash Player(10.3 版本)在视频及基础 Flash 动画的处理方面表现不俗,大家可以旋转视角并通过热点打开相关内容,而 Flash 游戏则时灵时不灵。相比之下,Windows Phone 7 完全不支持 Flash。

**优胜者 :** Android 2.3, 它的胜出得益于更强大的 HTML 5 兼容性 & 更好的个人信息控制。如果对 Flash 的支持对大家来说很重要,那么 Android 将是我们惟一的选择。

### Windows Phone Mango 对 Android 2.3: 用户界面

说起 UI 界面,Windows Phone 与 Android 系统间有明显的不同。Windows Phone 中的 Metro UI 清新、优雅、简单而且极具魅力——用户很快就能搞清楚如何在其中实现常用功能。相比之下 Android 2.3 的 UI 混乱不堪。用的时间越久,对它的抱怨就越多。虽然 Android 4.0 承诺要对此进行大规模修整,但根据谷歌所放出的演示来看,新 UI 在外观上也许更漂亮、风格上可能更统

一,但仍然不容易使用。但 Windows Phone 的简单 UI 同时也暴露了其功能上的简陋;比方说,我真的很难想象 Mango 的 UI 能够处理复杂的、多层次的互动操作。

Windows Phone 在操作方面坚持统一的风格——向右划动以获取更多内容,而点击“更多”按钮可以查看当前未被显示出来的其它功能。Android 的操作则更多地依赖于硬件及图标按钮,对用手势实现功能则不太关注,所以完成操作的过程同时也是不断点取对手指的折磨过程。

Android 手机虽然具备“搜索”按钮,但它并不是总能正确发挥作用。不过,如果我们在查看联系人时按下,就会启用电话簿查询功能。目前尚不清楚为什么“搜索”键在某些情况下生效,而其它情况下则没反应,特别是在类似电子邮件这类应该配备搜索功能的应用程序中。而相比之下,Windows Phone 的“搜索”按钮无论在何时按下都只会发生一件事: 打开 Bing 搜索应用。尽管统一性比 Android 更好,但我还是得抱怨一下 Windows Phone UI 中的两大白痴问题:

◆ Windows Phone 一贯采用锐利细小的字体,以致于大部分成年人都没办法在不戴老花镜的情况下进行阅读。

◆ Windows Phone 的版块及列表在填充屏幕方面表现神速,而这一点在用户垂直滚动较长的页面时就变成了累赘。

**优胜者 :** Windows Phone Mango——当然前提是大家未满 30 岁或者是有一副舒适的老花镜。Mango 及其应用程序在操控上要胜过 Android 系统,特别是如果大家平时常用的应用程序较少。



## 美女与极客之战 :Mango VS Android III

### Windows Phone Mango 对 Android 2.3: 安全性与管理

正如我在文章开头所说, Mango 缺乏大型业务运作所必需的安全管理能力。它所能执行的少数 Exchange 规程包括需要输入密码才能使用设备、要求密码具备一定复杂性、密码到期更换机制、防止密码重复使用并且在设备丢失或被盗时允许对内容进行远程擦除等。但其中没有 VPN 支持及设备加密功能,而这两种正是典型的企业级业务需求。

但是 Android 2.3 的密码管理控制机制比 Windows Phone Mango 还要少。不过摩托罗拉移动向谷歌提供了与 iOS 标准比较接近的安全解决方案。而且我们可以使用第三方应用程序为 Android 提供安全性与管理空间。

与 Windows Phone 不同, Android 支持 VPN。不过 Android 2.3 可以非常轻松地连接具备许可验证机制的无线局域网,但是在这方面 Windows Phone Mango 则完成得轻松加愉快。

Android 系统能够将设定、联系人、日程安排以及电子邮件数据以无线的形式备份到我们的谷歌账户中。而 Windows Phone 则没有这类功能。不过后者却能够在手机被遗失时,帮我们从微软的网站上找回失物——只需我们在该设备上曾经输入过 Windows Live 帐户信息即可。Android 系统中没有此类标准化功能,但在某些 Android 设备制造商处用户仍然能够获得同样的服务。

**优胜者:** 平手,或者说双输,因为二者在这方面功能中都相当薄弱。无论如何, Android 系统可以通过第三方软件的辅助带来更好的安全

性,而 Windows Phone 则不行。

### Windows Phone Mango 对 Android 2.3: 总体优胜者

当一切内容都分析过也讨论过,那么美女到底有没有战胜极客呢?从绝对数量来说,没有——从这个角度来看 Android 系统甚至强过 Windows Phone。不过当这个问题延伸至个人时——也许这位用户根本不打算在什么智能手机上瞎耽误工夫——那么 Windows Phone 的优势则极为明显,毕竟简单就是美嘛。

原文链接: <http://mobile.51cto.com/hot-299614.htm>

#### 扩展阅读

51CTO 微软 Mango 开发专题

<http://mobile.51cto.com/windows-phone-264461.htm>



51CTO 谷歌 Android 4.0 开发专题

<http://mobile.51cto.com/android-297987.htm>



# 李洋专栏：Android应用签名概述

作者 / 李洋

在 Android 系统中,所有安装到系统的应用程序都必有一个数字证书,此数字证书用于标识应用程序的作者和应用程序之间建立信任关系。

## 为什么要签名

应用程序签名是一个这样的过程,即使用私有密钥数字地签署一个给定的应用程序,以便达到如下 3 个目的:

- ◆ 识别代码的作者
- ◆ 检测应用程序是否发生了改变
- ◆ 在应用程序之间建立信任

基于这一信任关系,应用程序可以安全地共享代码和数据。如果一个 permission 的保护级别为 signature,那么就只有那些跟该 permission 所在的程序拥有同一个数字证书的应用程序才能取得该权限。Android 使用 Java 的数字证书相关的机制来给 APK 加盖数字证书,要理解 android 的数字证书,需要先了解以下数字证书的概念和 java 的数字证书机制。Android 系统要求每一个安装到系统的应用程序都是经过数字证书签名的,数字证书的私钥则保存在程序开发者的手中。也就是说如果一个 Android 应用程序没有经过数字签名,是没有办法安装到系统中的!Android 通过数字签名来标识应用程序的作者和在应用程序之间建立信任关系,不是用来决定最终用户可以安装哪些应用程序。这个数字签名由应用程序的作者完成,并不需要权威的数字证书签名机构认证,它只是用来让应用程序包自我认证的。

尤其值得注意的是:与信息安全领域其他使

用数字证书的用途不同,Android 将数字证书用来标识应用程序的作者和在应用程序之间建立信任关系,不是用来决定最终用户可以安装哪些应用程序。这个数字证书并不需要权威的数字证书签名机构认证,它只是用来让应用程序包自我认证的。也就是说,这个证书完全是由开发者来进行控制和使用的。

我们建议同一个开发者的多个程序尽可能使用同一个数字证书,这可以带来以下好处:

◆ 有利于程序升级,当新版程序和旧版程序的数字证书相同时,Android 系统才会认为这两个程序是同一个程序的不同版本。如果新版程序和旧版程序的数字证书不相同,则 Android 系统认为他们是不同的程序,并产生冲突,会要求新程序更改包名。

◆ 有利于程序的模块化设计和开发。Android 系统允许拥有同一个数字签名的程序运行在一个进程中,Android 程序会将他们视为同一个程序。所以开发者可以将自己的程序分模块开发,而用户只需要在需要的时候下载适当的模块。

◆ 可以通过权限(permission)的方式在多个程序间共享数据和代码。Android 提供了基于数字证书的权限赋予机制,应用程序可以和其他程序共享该功能,把数据给那些与自己拥有相同数字证书的程序。如果某个权限(permission)的保护级别是 signature,则这个权限就只能授予那些跟该权限所在的包拥有同一个数字证书的程序。

## 李洋专栏: Android 应用签名概述 II

### 签名的相关问题

在签名时,需要考虑数字证书的有效期:

◆ 数字证书的有效期要包含程序的预计生命周期,一旦数字证书失效,持有该数字证书的程序将不能正常升级。

◆ 如果多个程序使用同一个数字证书,则该数字证书的有效期要包含所有程序的预计生命周期。

◆ Android Market 强制要求所有应用程序数字证书的有效期要持续到 2033 年 10 月 22 日以后。

另外,Android 数字证书还包含以下几个要点:

◆ 所有的应用程序都必须有数字证书,Android 系统不会安装一个没有数字证书的应用程序。

◆ Android 程序包使用的数字证书可以是自签名的,不需要一个权威的数字证书机构签名认证。

◆ 如果要正式发布一个 Android,必须使用一个合适的私钥生成的数字证书来给程序签名,而不能使用 `adt` 插件 或者 `ant` 工具 生成的调试证书来发布。

◆ 数字证书都是有有效期的,Android 只是在应用程序安装的时候才会检查证书的有效期。如果程序已经安装在系统中,即使证书过期也不会影响程序的正常功能。

◆ Android 使用标准的 java 工具 `Keytool` and `Jarsigner` 来生成数字证书,并给应用程序包签名。

### 两种模式

构建 Android 应用程序时可以采用调试模式和发布模式:

(1) 使用 Android 构建工具 ( 命令行和 Eclipse ADT) 构建的应用程序是用一个调试私有密钥自动签名的;这些应用程序被称为调试模式应用程序。调试模式应用程序用于测试,不能够发布。注意,未签名的或者使用调试私有密钥签名的应用程序不能够通过 Android Market 发布。在调试模式下,Android 的开发工具会在每次编译时使用调试用的数字证书给程序签名,开发者无须关心。

(2) 准备发布自己的应用程序时,开发者必须构建一个发布模式的版本,这意味着用私有密钥签署应用程序。当要发布程序时,开发者就需要使用自己的数字证书给 APK 包签名,可以有两种方法。

◆ 在命令行下使用 JDK 中的 `Keytool`( 用于生成数字证书 ) 和 `Jarsigner`( 用于使用数字证书签名 ) 来给 APK 包签名。

◆ 使用 ADT Export Wizard 进行签名 ( 如果没有数字证书可能需要生成数字证书 )。

原文链接: <http://mobile.51cto.com/hot-300209.htm>



更多李洋专栏文章请点击:

<http://mobile.51cto.com/netsecurity-292955.htm>

# 杨亨冉Silverlight实现跨平台游戏开发

相对于 iOS 和 Android 的开发, Windows Phone 开发还是有它独特的优势的。比如,它很好的跟微软开发工具结合起来,以及与微软成体系的产品都能很好的衔接。你可以很容易的把你用 Silverlight 开发的网页应用移植到 Windows Phone 手机上。



中游在线就正在做这样的事。作为微软的 MVP, 中游在线的总经理杨亨冉对 Silverlight 技术非常熟悉, 中游在线公司所有的游戏都是利用 Silverlight 来开发的, 包括大型的 MMORPG 和各种休闲类的小游戏。同时, 他们还积极致力于 Silverlight 技术的推广, 包括推出了一些开发工具以及定期举办线下讨论沙龙等。



## Silverlight 跨网页与移动平台的移植

在回答 51CTO 记者“将 Silverlight 开发的网页游戏移植到 Windows Phone 手机上有多大难度”的问题时, 杨亨冉很轻松的说“几乎没有什么难度”。他表示, 移植中更多的精力会放在对 UI 适配的解决上, 毕竟大屏幕的网页游戏与小屏的移动终端在视觉体验上会很不同。此外, 还有一些代码上的细节问题, 比如, 网页版游戏你需要用工程模板来实现类似前进后退这样的导航功能, 但是在 Windows Phone 7 上, 由于系统本身就带

有翻页的功能, 所以移植的时候要把原代码中跳转的部分去掉。

“代码上的话, 除非你用了很多 Silverlight 网页上一些新的特性, 或者 Windows Phone 7 上很多特性, 改起来就麻烦一点”, 杨亨冉说。

在去年的 TechEd 上, 杨亨冉曾现场用 Silverlight 写了一个简单的游戏, 然后把它移植到 Windows Phone 7 上, 整个过程不超过 15 分钟。而对像《窝窝世界》这样大型 MMORPG 游戏的底层移植, 只用了不到两个星期。

当然, 这其中有个细节是, 《窝窝世界》的移植是在 Windows Phone Mango 版上进行的。因为最初的 Windows Phone 7 版本不支持 Socket, 在 7.1 即 Mango 版新增了这一特性。

不久之前, 有报道称, 微软将在 Windows 8 上推出基于 Silverlight 的应用商店, 这样一来, 开发者使用 Silverlight 开发的应用就同样可以发布在 Windows 8 系统。“Windows 8 针对平板开发的系统, 当然你也可以把它用在 PC 上”, 杨亨冉评论道, 他认为将 Market palce 放回 Windows 8 上将有可能改变中国软件销售行业的格局, 因为这样以来一些传统的靠推荐 windows 软件生存的网站可能就没什么意义了, 此外, 微软这一举动将可能有效杜绝盗版问题的存在。



## 杨亨冉:Silverlight 实现跨平台游戏开发 II

### 对 Silverlight 初学者的忠告

既然 Silverlight 有这样多的好处,而且微软显然将其放到了一个比较重要的位置,那么对于一个开发者来说,如果想涉足 Silverlight 开发,需要哪些储备呢?

杨亨冉说,如果是一个从来没有接触过 Visual Studio 相关开发的人员,他只需要安装 Visual Studio, Express 版本(学习版)和 Silverlight Tools for Visual Studio 基本就可以满足开发需求这样 coding 的环境基本就完成了。

设计的部分,用微软的 Expression Blend, Design, Web 来完成,他们的功能分别对应我们比较熟悉的 Flash, Photoshop 和 Dreamweaver。

这套工具的特点就是有效提升了开发速度。杨亨冉举了一个例子:有一次,运营商要在《窝窝世界》中加一个摆摊系统,开发团队用了不到一天就完成了这个功能,而如果用传统的方法,通过 PS 设计、前端制作、代码配合一整套流程下来,通常最短需要四天左右才能完成。

不过,对于一些没有接触过微软产品的开发者来说,开始可能会遇到一些问题。这主要是由于不同开发语言的思维方式不同。比如,非 .NET 体系的开发者可能对事件体系不太适应。

小的方面,杨亨冉举例说,初学开发人员可能经常会遇到一个问题是“跨线程访问无效”。这是因为 Silverlight 事件调用的时候会调一个 UI 线程,如果另外一个事件或线程操作了 UI 线程中的一部分,就会出现这种情况。解决方法也很简单,你需要做一个线程队列梳理清楚,然后在合适的时机调用它。

### 介绍游戏运营经验

中国的游戏开发者经常遇到的问题是,在技术和创意方面做的还不错,但是运营推广比较薄弱。作为在游戏领域有着多年开发、策划、运营经验的专家,杨亨冉有什么样的建议呢?杨亨冉认为,一个好的游戏需要具备几个要素:第一用户感官上、过程中是一种享受。也就是游戏界面要符合用户体验,故事逻辑要有剧情的深度,要有一种文化在里面。其次,收费点要合理,大家都喜欢的,游戏的寿命会很长。最重要的也是最基础的,游戏要稳定,bug 越多用户流失的越严重。此外,不同平台的游戏在设置考虑上也要有所区别。移动设备上的游戏主要占用碎片时间,因此目前一些休闲类的小游戏比较受欢迎,大型游戏要在移动端发展,需要设置一些特别的点,吸引用户随时回头。杨亨冉认为将来移动端和 PC 端肯定是融合的趋势,同一个游戏,用户可能在不同的场景用不同的终端完成不同的事情。

针对目前国内游戏界“重技术轻策划”的现象,杨亨冉认为,重技术是因为技术普及的太慢,像 Silverlight 推广就很难,技术形成需求瓶颈后,大家于是都追求技术。但是这一现状能够在不久的将来得到改观,毕竟“技术目的是为产品服务,技术的盘子有多大,产品就能做多大”。

谈到 HTML5 和 Silverlight 谁更有前景的问题,杨亨冉表示“HTML5 是现在的趋势,但不是现在的优势”。优势可以变成收益,但趋势不能。厂商瞄准 HTML5 是因为它是将来互联网很重要的领域,谁站稳了谁就有可能成为互联网新霸主,但是现在 HTML5 还是不是一个完善的标准。

# 挑战者诺基亚：放下辉煌，从头再来

10月26日，在诺基亚世界大会上，我们终于盼到了传闻已久的诺基亚 Windows Phone 手机正式发布。此时距离诺基亚宣布与微软结盟的消息已经过了8个月。时间太短了，短到让人怀疑诺基亚是否已做好调整以进行新的战斗。时间又太长了，长到对手在不到一年的时间里连创佳绩将差距越拉越大。

越王勾践可以卧薪尝胆十年以图东山再起，但瞬息万变的移动互联网市场，可等不起十年二十年的修炼，诺基亚从颓势显露到让出王位再到如今携微软重打擂台，整个过程也不过用了两三年的时间。

黑马崛起的剧情固然引人注目，王者归来的桥段却更加惊心动魄。然而，诺基亚这个昔日的王者，究竟能否载誉而归呢？



## 第一次反击：差强人意

诺基亚在本次世界大会上一举发布了两款 Windows Phone 智能手机：Lumia 800、Lumia 710。

其中 Lumia 800 拥有与 N9 极为相似的外观，采用 3.7 英寸 AMOLED 电容式触控屏，高通 MSM8255 1.4GHz 单核处理器及 16GB 内置存储容量，512MB RAM，搭载一枚 800 万像素摄像头，预计零售价为 420 欧元。仅从配置和价格上来说，

新机型并无特别的亮点。以致大多数观众都显得有些失望。定位较高端的 Lumia 800，竟然采用了与 N9 极为相似的外观，真让人百感交集。

由于 MeeGo 系统的前途未卜，N9 的前景实际并不怎么光明，然而，这款机型无论外观还是处理性能，都可圈可点，在一些开发者和发烧友那里得到了极高的评价，以至于在得知 Lumia 800 采用了 N9 设计之后，一位网友评价道：“诺基亚牺牲了自家的 N9！”除了让出了 N9 备受称赞的外观设计，在软件上，诺基亚也不遗余力的倾注到这两款手机上来。两款手机搭载了诺基亚地图、诺基亚 Drive、运动 Hub、Mix Radio、诺基亚音乐软件等诺基亚自身的软件产品，以示与其他 Windows Phone 厂商的区别。不过，仅有这些是不够的，Windows Phone 自身生态链的健康发展才是诺基亚 Windows Phone 手机的坚实后盾。

据媒体报道，微软 Windows Phone 的应用数量最近突破了 3.5 万，而与此同时，App Store 应用数量为 60 万，Android Market 也已达到 50 万。这样的量级着实让人替诺基亚和微软担心。不过，两家厂商也不会坐以待毙，不久前的 TechED 上微软明确表示将与开发者加强合作，诺基亚也宣布将提供上百万部 Windows Phone 手机给开发者免费使用。这些措施的效果如何，Windows Phone 应用会不会迅速爆发，我们还不得而知。

## 挑战者诺基亚：放下辉煌，从头再来 II

不管怎么说，战争终于正式打响了，诺基亚开始了它的“第一次反击”。10月26日当天，诺基亚股价在纽约证券交易所常规交易中上涨0.06美元，涨幅0.89%，报收6.77美元，也算是资本市场对次反击的小小鼓励吧。

### 放下辉煌，从头再来

在对未来移动互联市场的格局猜测中，有一种观点非常普遍，那就是大多数人认为未来会是苹果、谷歌、微软三分天下的局面。对，这其中没有诺基亚。

诺基亚躲在微软背后。在注重生态圈结构的移动互联网时代，失去了操作系统的诺基亚很难再站上辉煌的前台。诺基亚与微软结盟之前，外界对诺基亚会否投入Android阵营曾有诸多猜测，最后，诺基亚选择了微软。原因之一是，投入Android阵营有可能会使诺基亚沦为“纯粹的硬件提供商”，但与实力相对较弱的微软结盟，则多了许多议价条件。而且，微软有“Wintel”的历史，恐怕不会有谷歌软硬合一的野心，但未来谁能说的定呢？

谷歌收购摩托罗拉之后，我们听到最多的猜测就是：微软会收购诺基亚。

市场非常看好这一前景。但诺基亚自己恐怕还不大乐意。

那么，翻身，诺基亚还有机会吗？

我们引用诺基亚研究院的一名研究员的观点，他分析说，要让诺基亚重现当年的辉煌，5年内是不可能了，但是如果以扭转智能手机市场份额持续下滑的颓势为目标，则有一定可能。该研究员表示，如果诺基亚能将塞班服务做好，守住中

低端用户群，等待Windows Phone日渐成熟慢慢完成过渡，未来或许还有机会。该研究员评论说“诺基亚目前的战略是高端靠Windows Phone顶住，少输当赢，重塑品牌形象，靠中低端抢占市场份额。”

这话听起来有几分道理。中低端用户的确占诺基亚销量的很大一部分，这也是时至今日诺基亚仍能保持一定市场份额的中坚力量。不过，随着智能手机的日渐普及，以及市场舆论导向的引导，这部分用户很有可能投靠Android智能手机阵营——还记得去年成批成批出现的“Android千元手机”吗？一位分析师曾评论说“如果Android手机价位降到500元左右，诺基亚就完全没有机会了”。

对于诺基亚来说，选择被微软收购，前路也许会顺畅的多，只是要永远“听令与人”；如果仍坚持独立拼杀，它将面临两个问题：其一，与微软的关系如何维护，如何在众多Windows Phone手机中保持优势？其二，在走全生态系统发展的市场格局下，是否还要继续“染指”操作系统？

我们并不清楚诺基亚内部正在做怎样的考量，不过，至少诺基亚已经看清了态势，在积极寻找解决方案。在不久前的中国移动开发者大会上，来自诺基亚开发者生态系统及体验的大中国区总监黄耀丰将诺基亚称为“挑战者”——我们很高兴看到这样的心态，放下昔日辉煌，以一种空杯的心态，从头再来。

原文链接：

<http://mobile.51cto.com/news-300600.htm>

## ■ 编者按

在这个百年一遇的“光棍节”(2011年11月11日)来临之前, Adobe“抛弃”了移动版的Flash player: 其宣布将停止开发移动版和TV版的Flash插件。

## Adobe的光棍节:与移动版Flash player离婚之后

2011年11月9日, Adobe“抛弃”了移动版的Flash player: 宣布将停止开发移动版和TV版的Flash插件。这件事情被一些媒体解读为“Adobe放弃了移动版的Flash”, 然后我们看到很多读者表示“当时就被震惊了”。

事实上, Adobe公司平台副总裁兼总经理丹尼·维诺科(Danny Winokur)在博客中的原话是这样的:“今后, 移动Flash的业务主要是帮助开发者用Adobe AIR打包应用。在面向Android和BlackBerry PlayBook的Flash Player 11.1推出后, 移动版Flash Player将停止开发。我们会继续为当前设备提供重要补丁和安全更新, 源代码也会继续有效。”

也就是说, Adobe并没有彻底放弃移动版Flash, 它只是以另一种形式存在了。一位用户在微博上发布的这张图片清晰的解释了Adobe新宣布的Flash和HTML5策略。



### Adobe 为何这么做?

诚如 Adobe 公司所解释, HTML5 已经成为

建立移动浏览器内容的优先标准, 而此前 Adobe 在移动终端浏览器 Flash 插件上的发展并不太成功, 与其无谓的浪费资源不如选择一种更有前景的工作方式。而且, 在这个过程中, Adobe 其实并没有很大的损失。

一方面, 在移动设备上, App 应用程序已经越来越多的取代网页浏览, 引用一位网友的评论说:“Flash 能做的, 除了在浏览器中运行之外, AIR 都能做; AIR 能做的, Flash 并不是都能做。既然在移动设备中, 原生程序才是王道, 那何苦还要开发费力不讨好的 Flash 呢?”

另一方面, 丹尼·维诺科在博客中表示:“通过调整, 我们可以加大对 HTML5 和 Flash 创新的投入。我们还会继续利用 Flash 开发经验推动 W3C 和 WebKit 合作, 尽快在 HTML5 上提供类似功能。我们会把 Flash 新功能转移到 HTML5”——表面上看这一回合在移动终端浏览器上的争斗 Flash 是“输给”HTML5, 但 Adobe 参与到 HTML5 的工具开发中, 这即是对新技术的拥抱, 也是 Flash 的另一种“重生”。

此外, 放弃没有前景的移动终端浏览器 Flash 插件, 也让 Adobe 的产品战略更加聚焦, 可以集中精力让 Flash 为 PC 用户提供更好的网页浏览、游戏和视频体验。Adobe 称“正在开发 Flash Player 12 会带来高清娱乐体验”。



## Adobe 的光棍节：与移动版 Flash player “离婚”之后 II

尽管 Flash 不是 Adobe 最赚钱的产品——花旗银行分析师估计 2009 年 Flash 只占 Adobe 营收的 7%。然而,有分析师认为 Adobe 此举放弃的是一条潜在的产业链:如果移动版 Flash 能获得足够高的份额,Adobe 不仅可以通过授权提升自己的收入,未来还可以从事广告、内容分发、数据分析等业务。——考虑到 Adobe 并没有真正的放弃移动产品,笔者并不认同这一观点。当然,AIR 的推广也面临难度,其虚拟机的下载安装会是一个推广的门槛。在一段时期内,这一举措会对 Adobe 在移动互联网的占有率造成一定影响。

### 业界反应：苹果笑，Android 哭？

在宣布不支持移动版 Flash Player 之前,Adobe 刚刚放出重组和裁员消息。Adobe 表示将重组公司业务,在北美和欧洲裁员 750 人。也市场得知裁员和不支持移动版 Flash Player 之后,Adobe 股价严重受挫,盘前下跌超过 10%。

比股价下跌更可怕的是业内几乎一片“**挺苹果,倒 Adobe**”的声音。去年四月份,乔布斯在美国发表声明,指责 Flash 技术漏洞百出,而且程序消耗电池,不适合安装在 iPhone 及 iPad 中。乔布斯取笑 Flash 是一款没有未来的产品。

在被媒体渲染过的“移动版 Flash Player”停止更新的消息传出后,51CTO 记者看到微博上、论坛上大部分人都在称赞乔布斯“又一次准确预测了未来”,人们纷纷向乔帮主致敬,向 HTML5 致敬,俨然 Adobe 就要完了的架势。果粉们嘲笑 Android 用户,从此以后再也没有什么优势值得夸耀了——此前,iPhone 不支持 Flash 而 Android 支

持,这曾是 Android 用户最引以为豪的事情。目前,谷歌方面还没有对此发表评论,不过同样在移动终端支持 Flash 的 RIM 已经表示作为 Flash 的部分源码持有人,将在其 PlayBook 平板上为客户提供 Adobe Flash Player 的原生支持,在黑莓浏览器上也同时提供 HTML5 和 Flash 的支持。

部分 Flash 开发者表示忧心忡忡,不知道路在何方。不过,也仍有一些 Flash 爱好者对前景充满信心。其中 Adobe 平台技术经理李鹏应该是 Flash 最大的粉丝。Adobe 已经通知他明年将会把他所负责的“Flash Pro 传教士岗位”岗位撤销,李鹏表示如果将来公司安排的工作与 Flash 无关,他将离开 Adobe。李鹏在微博上称“以我对 Flash 的认识,我坚信它在相当长一段时间内还是无法替代的主流前端技术”。的确,不管在移动端的发展情况如何,起码在 PC 端,Flash 有深厚的用户基础,这是不容易撼动的。而且,即便 HTML5 技术会有一定威胁,但威胁来临也需要一些时间。

论坛上一位用户表示:Adobe 今天可以放弃移动版 Flash Player,未来某一天,当移动端和 PC 端的用户体验统一起来,也许那时,Adobe 放弃的就是整个 Flash 了——我觉得其实我们还可以考虑另一种可能性,那时不断发展的 PC 端 Flash 会完美移植到移动端,或者 Adobe AIR 将会更加完美的完成多终端支持。

退一万步说,就算 Flash 哪天寿终正寝,那也是正常的新旧更迭,作为科技公司,重要的不是死守着一款产品,而是要不断改进、甚至推翻原有产品。毕竟“技术没有成败,只有不断向前发展”。



### 【活动主题】

重温“TechEd” Windows Phone 开发经典课程

### 【演讲嘉宾】

“TechEd2011” Windows Phone 开发课程讲师：李靖南,王涛,杨亨冉,马宁

### 【活动亮点】

本期 Phone Club 带您重温 “TechEd2011” Windows Phone 经典课程,4 位专家为 “TechEd2011” Windows Phone 开发课程讲师,并有 Windows Phone 手机作为抽奖礼品。

【活动报名地址】 <http://mobile.51cto.com/mobile/phoneclub/>

### 【活动时间】

2011 年 12 月 17 日 13:00-17:00

### 【活动地点】

北京市海淀区双清路学研大厦 2 层多功能厅。

公交车：333 路、355 路、438 路、466 路、628 路、913 路和运通 110 路清华东路西口站下车,沿双清路向东北步行 260 米。



【报名人数】限 200 人

【报名截止】2011 年 12 月 16 日

【联系我们】[phoneclub@51cto.com](mailto:phoneclub@51cto.com)